IBM Maintenance Library

3725/3726 Communication Controller and Expansion Diagnostic Descriptions

SY33-2027-4

Fourth Edition (June 1986)

This is a major revision of SY33-2027-3, which is now obsolete. Changes to the text or illustrations are indicated by a vertical line to the left of the change. This edition reflects the 3725 Release 4 enhancements.

PREFACE

This publication is intended for the customer engineer who maintains the IBM 3725 Models 1 and 2 Communication Controllers and the IBM 3726 Communication Controller Expansion. It complements the 3725/3726 Maintenance Information Manual (MIM) Part 1, Volumes 1 and 2, (SY33-2018) and the part-numbered IBM 3725/3726 MIM Part 2 by describing the Diagnostic Programs used with the IBM 3725/3726.

The reader should be trained on the IBM 3725/3726, and have an understanding of telecommunications and modems. The Introduction to the IBM 3725 Communication Controller, GA33-0010, is a prerequisite publication.

The 3725 Communication Controller, Operating Guide, GA33-0014 and the 3725 Communication Controller, Problem Determination and Extended Services, GA33-0014 are corequisite manuals that provide the procedures available for operating the communication controller.

SUMMARY OF CONTENTS

A set of separators (SY33-2028) is available.

BIBLIOGRAPHY

The bibliography lists the publications that describe the IBM 3725 Communication Controller and its software support. The publications are grouped into introductory, planning and installation, operating, and maintenance.

MAINTENANCE LIBRARY ORGANIZATION

Identifier	Title	Contents
MI Vol. A01	MIM Part 2 "START"	Start of problem isolation
MI Vol. A02	Power Supplies	MIM Part 2, power MAPs and power supplies theory of operation
MI Vol. A03	MIM Part 1, Vol. 1	Tutorial and general information
MI Vol. A04	MIM Part 1, Vol. 2	Unit descriptions
MI Vol. A05	Diagnostic Descriptions	Diagnostic information
MI Vol. A06	Operating Guide Wrap Tests Stand-Alone Link Tests	Operating information for customer and service personnel
MI Vol. A07	Parts Catalog Installation Instructions	Illustrated parts catalog Field engineering installation instructions
MD Vol. B01	Component Locations Wiring Diagrams	EC controlled information in support of MIM Parts 1 and 2
MD Vol. B02	Pin Lists Net Lists	

CONTENTS

CHAPTER 1. DIAGNOSTIC PROGRAMS

Table of Contents for Chapter 1

CHAPTER 1. DIAGNOSTIC PROGRAMS

WARNING	When you are running the diagnostic programs, the customer cannot use the 3725.

The diagnostic programs are run to detect solid failures caused by the hardware in the 3725, and to isolate the field replaceable unit that caused the failure. They are also run after a repair is performed to check that the controller is working correctly. Diagnostics must be run before and after an EC or an MES is installed.

Only the channel adapters, scanners, and telecommunication lines defined in the 3725/3726 configuration data file are tested. Run the CDF 'verify' option when you suspect a discrepancy between the machine configuration and the CDF (see page 2-401 of the MIM Part 1).

A repair action code (RAC) is displayed on the console screen when a diagnostic program detects a failure. Refer to MIM Part 2 for handling this RAC.

Repair: With the exception of the 3727 console and the diskette drive, all repairs are performed with the 3725 offline.

Diagnostics consist of:

1. Channel adapter OLTs stored in the host, and the OLT responder stored on the service diskette.
2. ST370, NST-2, and ST4300 (system tests). For details, see page 2-160 of the MIM Part 1.
3. IML checkout programs stored in read-only storage (ROS). For details of these programs, see "Initialization" in Chapter 6 of the MIM Part 1.
4. Offline diagnostics stored on the service diskette.

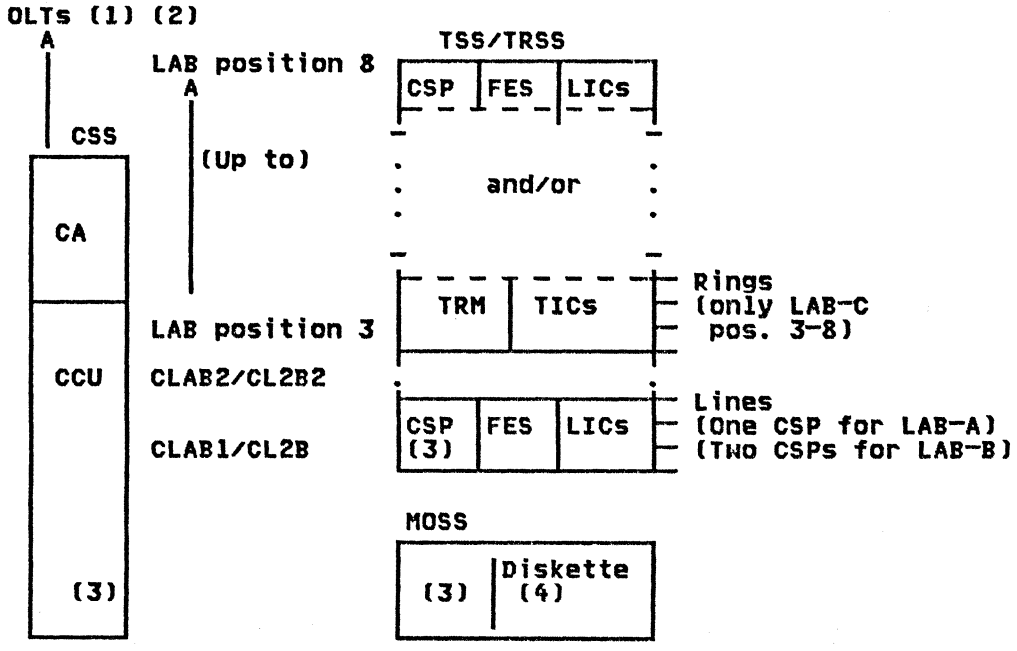


Figure 1-1. Diagnostic Testing Sequence

OFFLINE DIAGNOSTICS

Once the MOSS is initialized with its microcode, the offline diagnostics can be run and the isolation of a failing FRU can start.

The offline diagnostics are monitored by the diagnostic control monitor and the command processor.

DIAGNOSTIC CONTROL MONITOR (DCM)

The diagnostic control monitor is loaded by selecting 'D' from the primary menu of the service diskette. It automatically restricts the diagnostic testing to an element defined in the configuration data file (CDF).

Communication with the DCM is through the 3727 console. The DCM allows diagnostic program selection and choice of options within the selection. It sends your commands to the command processor, and displays diagnostic results on the console.

COMMAND PROCESSOR (CP)

The command processor is loaded in the 3725 subsystem where the selected diagnostic is to be run. It passes on the diagnostic commands such as 'start routine' and 'restart on error'. It also reports diagnostic events back to the DCM; these events can be routine end, routine service request, and diagnostic results.

DIAGNOSTIC CONTROL FACILITIES (DCF)

The DCM and the CP together provide a set of facilities for running the diagnostics, which are collectively referred to as the diagnostic control facilities (DCF) throughout this manual.

The following figure shows the diagnostic architecture. Arrows indicate logical communication between elements.

DCF Architecture

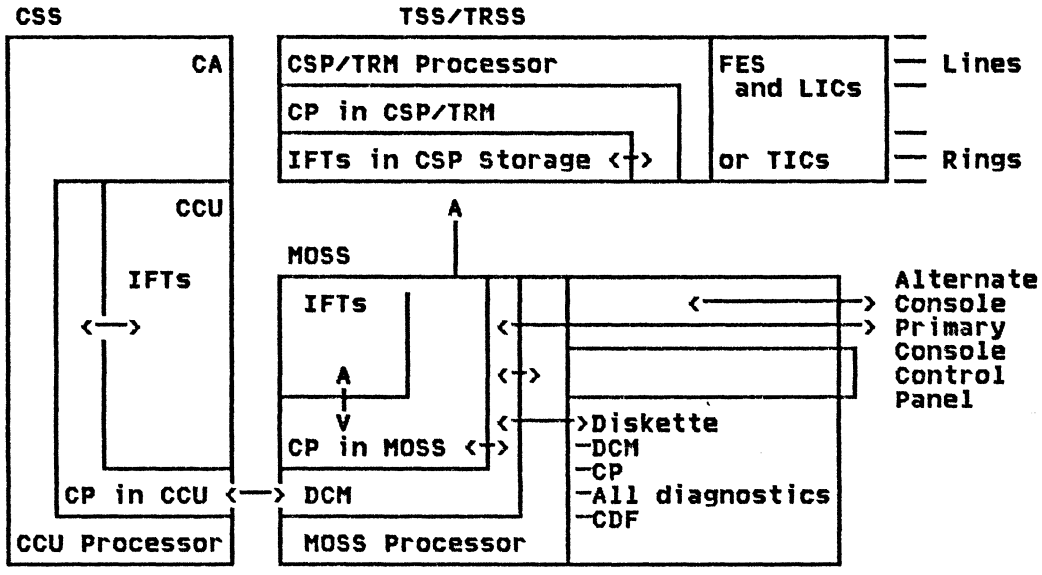


Figure 1-2. DCF Architecture

DIAGNOSTIC STRUCTURE

The offline diagnostics are arranged in groups, internal functional tests (IFTs), sections, and routines.

A **group** is a set of IFTs that tests a 3725 subsystem (the TSS group for example).

IFTs are often divided into **sections** that can be loaded and executed one at a time.

A **section** is a set of routines that tests a particular adapter, or a component of a subsystem.

The shortest executable test is the **routine.**

Diag Name	IFT	Residence	Run In
CCU (2)	A-F	Disk	MOSS + CCU
CA IFT(4)	L	Disk	CCU
CA OLT(6)	M	Disk + Host	CCU + Host
RDV(3)	I	Disk	CCU
IOCB(3)	J	ROS + Disk	CCU + CSP
CSP		ROS + Disk	CSP
FES(5)	P	Disk	CSP
LIC(5)	Q	Disk	CSP
ICC(5)	R	Disk	CSP
MPC		ROS + Disk	MOSS
MCC/MPC		ROS + Disk	MOSS
CCA/EIA		Disk	MOSS
TRA(7)	T	Disk	TRA

* The diagnostic group number is shown in parentheses. Group 1 includes all IFTs.

DIAGNOSTIC RUNNING SEQUENCE

Testing the 3725 with the off-line diagnostics assumes that MOSS and scanner IML is possible. When the option 'run all diagnostics' is selected, testing starts from the smallest element in a subsystem, and builds up step by step on error-free elements until a subsystem is completely tested. The diagnostics then continue with the other subsystems until the 3725 is completely tested.

The diagnostic running sequence must be followed. You must be sure when selecting a diagnostic group, IFT, or routine that the preceding testing works perfectly. If not, the results given by the diagnostic may be of no value, or worse, misleading.

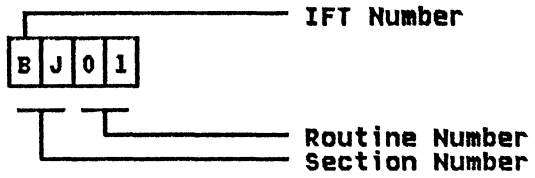
The running sequence starts from:

- Group 2 to group 7
- IFT A to Z, within a group
- Section A to Z, within an IFT
- Routine 01 to FF, within a section

For IFTs, sections, and routines available within a diagnostic group, refer to the diagnostic group descriptions that follow.

DIAGNOSTIC IDENTIFICATION

The identification contains the IFT number, the section number, and the routine number as follows:



For specific IFT, section, or routine selection, see 1-9.

Run Sequence	Mode of Control	IFTs (See note 1)
MOSS	(See Note 2)	
CCU	DCF	MIOC direct op MIOC indirect op Basic instruction Full instruction Full storage test Branch test/address compare
IOC BUS	DCF	RDV TRMs IOC bus (For details see Chapter 3) CSPs (ROS)
CA	DCF	Channel adapter
TSS	DCF	CSPs (ROS) FES LIC ICC
CA OLTs	DCF in machine OLTs in Host	OLT Responder (For details see Chapter 4)
TRSS	DCF	TRM TICs (For details see Chapter 7)

Notes:

1. For IFT running time, see each diagnostic group description in this chapter.

2. MOSS diagnostics are not part of the offline diagnostics. The MOSS is diagnosed while running:

   - MOSS IML using the service diskette and the Function Select switch on Maintenance.

   - MOSS IML using the controller diskette and the Function Select switch on Normal or MOSS IML. This second diagnostic is a subset of the first.

     For details of MOSS IML differences, see Chapter 6 in MIM Part 1 and Chapter 6 in this manual.

HOW TO RUN OFFLINE DIAGNOSTICS

The DCF can be selected from the control panel or from the operator console. The way in which diagnostics are selected does not affect their running.

Warning: Before entering the following procedure, set all Channel adapters 'disabled' and wait for 'all channel adapter disabled' (green light) on.

Procedure:

1. Mount the service diskette on the diskette drive.
2. Set the Function Select switch on the control panel to 'Maintenance'.
3. Press the Power ON Reset switch even though the 3725 is already powered on.

Note: When the service diskette is mounted, and the function select switch is in NORMAL position, attempts to initialize the 3725 result in a MOSS IML exception.

Resulting Screens: If starting from power-off, or if previous diagnostic runs have not needed the 'CCU Initialization' phase, the following screen is displayed:

```
                    SERVICE-MODE                    CCU INTERRUPTS DISABLED


D:DIAGNOSTICS                                SP:CCU STOP     Q:DATE/TIME
                                U:UTILITY PGM ST:CCU START    T:TERMINATE
E:ERROR LOG                                  RS:CCU RESET




===>
```

If previous diagnostic runs have needed the 'CCU Initialization' phase, the following screen is displayed:

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK

D:DIAGNOSTICS                                  SP:CCU STOP    Q:DATE/TIME
                                 U:UTILITY PGM ST:CCU START   T:TERMINATE
E:ERROR LOG                                    RS:CCU RESET

===>
```

Enter 'D'

Depending on whether one (or more) of the channel adapter interfaces is enabled, one of the following screens (diagnostic request menu) is displayed:

1. No Channel Adapter Interface Enabled:

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK

D:DIAGNOSTICS                                  SP:CCU STOP    Q:DATE/TIME
                                 U:UTILITY PGM ST:CCU START   T:TERMINATE
E:ERROR LOG                                    RS:CCU RESET

DIAG  |ADP# |LINE |
                  |
1 ALL             |
2 CCU             |
3 IOCB            |
4 CA  |1-> 6|     |
5 TSS |1->16|0->31|
6 OLT |1-> 6|     |
7 TRSS|6->16|1->4 |
      AND         |                                DIAG - RUN INIT
                  |
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                  | ENTER REQUEST ACCORDING TO THE DIAG MENU
                  | DIAG==>       ADP#==>     LINE==>     OPT==> N
===>
```

2. One or More Channel Adapter Interfaces Enabled:

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK


D:DIAGNOSTICS                                  SP:CCU STOP    Q:DATE/TIME
                                 U:UTILITY PGM ST:CCU START   T:TERMINATE
E:ERROR LOG                                    RS:CCU RESET

DIAG  |ADP# |LINE |
                  |
1 ALL             |
2 CCU             |
3 IOCB            |
4 CA  |1-> 6|     |
5 TSS |1->16|0->31|
6 OLT |1-> 6|     |
7 TRSS|6->16|1->4 |
       AND        |                                  DIAG - RUN INIT
                  |
OPT = Y IF MODIFY |
OPTION REQUIRED   | CAUTION: CA INTERFACE(S) ENABLED: A______ B____
                  | ENTER REQUEST ACCORDING TO THE DIAG MENU
                  | DIAG==>       ADP#==>     LINE==>     OPT==> N
===>
```

Warning: The 3725 may be operational; stop the 3725 correctly before starting the diagnostics.

SELECTING FROM DIAGNOSTIC REQUEST MENU

- Select from the menu the parameters you wish to enter on the selection line. Move the cursor from its initial position (DIAG==>) to the next after each parameter is entered. To skip a parameter entry, press the -->| key.
- Press SEND to execute the request.
- Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

Specific Selection

Any IFT, section, or routine can be specifically selected in the diagnostic selection area (DIAG==>). **Examples:**

1. To select the FES IFT P, enter:

   DIAG==> P

2. To select section A of FES IFT P, enter:

   DIAG==> PA

3. To select routine 10 of section A of FES IFT P, enter:

   DIAG==> PA10

Note: The manual routines are skipped during normal running of requests. You must specifically select such routines.

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK


D:DIAGNOSTICS                                    SP:CCU STOP    Q:DATE/TIME
                                  U:UTILITY PGM  ST:CCU START   T:TERMINATE
E:ERROR LOG                                      RS:CCU RESET

DIAG  |ADP# |LINE |
                  |
1 ALL             |
2 CCU             |
3 IOCB            |
4 CA  |1-> 6|     |
5 TSS |1->16|0->31|
6 OLT |1-> 6|     |
7 TRSS|6->16|1->4 |
       AND        |                                  DIAG - RUN INIT
                  |
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                    ENTER REQUEST ACCORDING TO THE DIAG MENU
                    DIAG==> PA10  ADP#==> 1   LINE==> 20  OPT==> N
===>
```

On this screen, the routine PA10 is selected. It will run on line address 20 (LINE==>20) of scanner number 1 (ADP#1).

Diagnostic Running Procedure

The menus available in the diagnostic mode are as follows:

- Diagnostic request
- Diagnostic selection modify
- Diagnostic execution modify
- Diagnostic message

The following figure shows which action on the console causes menu switching.

To stop a diagnostic program that is running, press the ATTN key.

Wait for the control program to receive the break caused by the ATTN key, and to stop the diagnostic (up to 3 minutes).

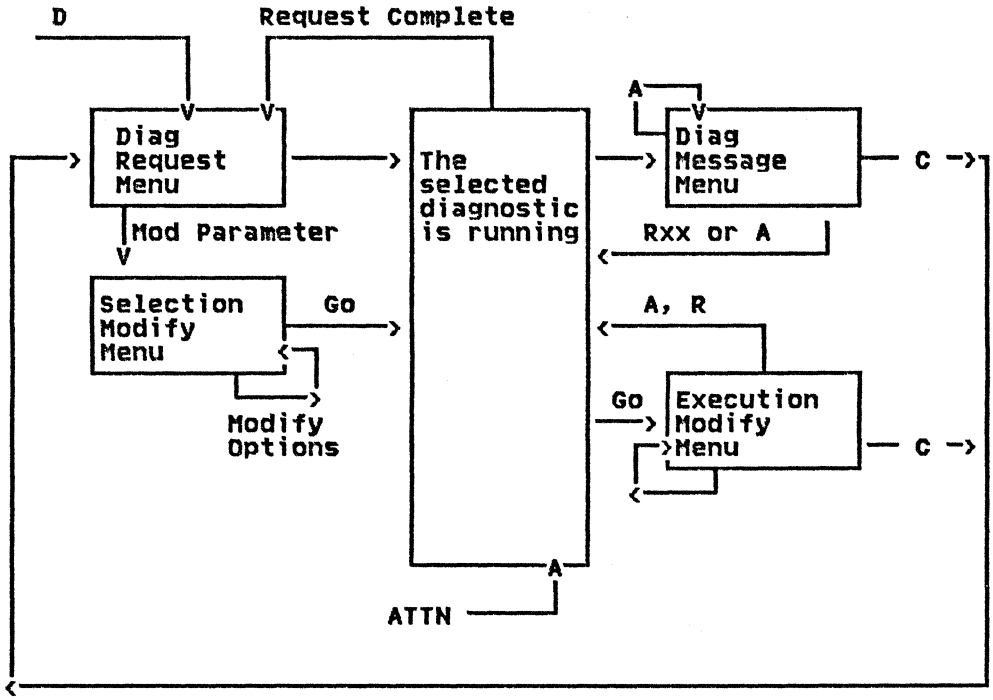


Figure 1-3. Diagnostic Running Procedure

USE OF CCU FUNCTIONS WITH DIAGNOSTICS

When running CCU diagnostics (IFTs A, B, C, D, E, and F), you cannot always use the CCU functions. If you try to use them when they are not allowed, the diagnostic menu displays: CCU FNCTN BARRED.

If you request 'RUN ALL', you may use the CCU functions **only after** IFTs A, B, C, D, E, F, I, and J have been run.

When running TSS diagnostics (IFT P, Q, and R) the CCU is not used as a processor.

Only the last 2K of CCU storage is used as 'scanner mailbox' for data exchange between the DCM in the MOSS and the CP-and-IFTs in the TSS. There is no need to use the CCU functions when running TSS diagnostics, except if you wish to display the 'scanner mailbox'. Setting an address compare, for example, has no meaning.

All CCU functions should be used only when running IFTs I, L, or M. All CCU functions except Branch Trace may also be used when running IFT J.

Instruction stepping is not supported by the DCF.

```
D:DIAGNOSTICS                                        SP:CCU STOP     Q:DATE/TIME
                                   U:UTILITY PGM     ST:CCU START    T:TERMINATE
E:ERROR LOG                                          RS:CCU RESET      CCU FNCTN

 1 ADDRESS COMPARE| - SELECT A SYSTEM CONTROL FUNCTION (1 TO 12)      ==> 11
 2 BRANCH TRACE   |      1 = SET I-STEP
 3 DISPLAY/ALTER  |      2 = RESET I-STEP
 4 DISPLAY LONG   |      3 = CCU LVL3 INTERRUPT
 5 SYSTEM CONTROL |      4 = CCU CHECK RESET
 6 DATA EXCHANGE  |      5 = SET BYPASS CCU CHECK
 7 AC/BT PARAM    |      6 = RESET BYPASS CCU CHECK
 8 CANCEL AC      |      7 = SET ADAPTER CHECK STOP
 9 CANCEL BT      |      8 = RESET ADAPTER CHECK STOP
10 CH ADAPT STATE |      9 = RESET IOC
11 RESET CCU/LSSD |      10= RESERVED
                  |      11= MOSS ONLINE
                  |      12= MOSS OFFLINE
                  |                              PF3:CCU START
                  | REQUEST IGNORED
===>
```

RESTRICTIONS

1. Using CCU functions 10 (CH ADAPT STATE) or 11 (RESET CCU/LSSD) may lead to unpredictable results.

2. In CCU function 5 (SYSTEM CONTROL), using function 3 (CCU LVL3 INTERRUPT) or 9 (RESET IOC) may lead to unpredictable results.

To use address compare and branch trace, the CCU must first be initialized. The setting of these two functions remains active as long as the CCU is not reinitialized (indicated by 'RESET' on the MSA).

Note: For branch trace, local store X'7D' **must** contain X'10000' (address of the branch trace buffer), and local store X'7C' **must** contain X'6000' (length of branch trace buffer). Check the local stores for these values, and set them to the correct values when required.

SELECTION

Proceed as follows:

- Enter your diagnostic request
- Enter G if an option is requested.
- Press the ATTN key to initiate a break.

  If the request is 'RUN ALL', do not press the ATTN key before the routine IA01 appears on the screen.
- After a short wait, you receive the 'BREAK RECEIVED' message.

  The CCU is now initialized.
- Enter the CCU function application by pressing the CCU key, then select the specific CCU function.
- Return to the DCF application by pressing the CCU FNCTN key again, then enter G (for go).
- You will regain control after an address compare has occurred. The following frame is displayed:

DCF Display on CCU Stop Address Compare

You may now use the CCU functions (use the CCU key to enter and to return to the DCF application).

```
PROCESS  STOP-CCU-CHK SERVICE-MODE AC HIT              LAR:...... OP:....   C:.
STOP-AC   BYP-ADP-CHK                                  IAR:...... ILVL:.    Z:.

D:DIAGNOSTICS                                    SP:CCU STOP    Q:DATE/TIME
                                  U:UTILITY PGM  ST:CCU START   T:TERMINATE
E:ERROR LOG                                      RS:CCU RESET

                        |
A ABORT ROUTINE         |
                        |
C CANCEL REQUEST        |
                        |
RX..X    REPLY:         |
 ENTER R                |
 FOLLOWED BY X..X       |
 (IFT MESSAGE),         | REQUEST: ....  .                DIAG RUNNING
 THEN PRESS SEND.       | OPTIONS: S   NW C1   R1         ROUTINE ....
 (REPLY MAY BE A        | REQUESTED CCU STOP OCCURRED.
 NULL STRING)           |
                        | ENTER REQUEST ACCORDING TO THE DIAG MENU
                        | ==>
===>
```

RESUMING DCF PROCESSING

To resume DCF processing requires two steps: START CCU, then restart the DCF.

Note: The second step (restart the DCF) is possible only when the CCU enters the run mode (see second line of MSA). One or several subsequent START CCU may be needed to move the CCU status to run mode, depending on the AC stop occurrences.

Any attempts to restart the DCF when the CCU is in STOP-AC mode may lead to unpredictable results.

1. Start the CCU by entering 'ST' as follows:

```
PROCESS  STOP-CCU-CHK SERVICE-MODE AC
RUN       BYP-ADP-CHK

D:DIAGNOSTICS                                SP:CCU STOP    Q:DATE/TIME
                                U:UTILITY PGM ST:CCU START   T:TERMINATE
E:ERROR LOG                                  RS:CCU RESET

                    |
                    |
A ABORT ROUTINE     |
                    |
C CANCEL REQUEST    |
                    |
RX..X     REPLY:    |
 ENTER R            |
 FOLLOWED BY X..X   |
 (IFT MESSAGE),     | REQUEST: ....  .                DIAG RUNNING
 THEN PRESS SEND.   | OPTIONS: S   NW C1   R1         ROUTINE ....
 (REPLY MAY BE A    | REQUESTED CCU STOP OCCURRED.
  NULL STRING)      |
                    | ENTER REQUEST ACCORDING TO THE DIAG MENU
                    | ==>
===> ST                                                START COMPLETED
```

2. Restart the DCF by entering 'R' (continue), 'A' (abort), or 'C' (cancel):

```
PROCESS  STOP-CCU-CHK SERVICE-MODE AC
RUN       BYP-ADP-CHK

D:DIAGNOSTICS                                  SP:CCU STOP    Q:DATE/TIME
                                 U:UTILITY PGM ST:CCU START   T:TERMINATE
E:ERROR LOG                                    RS:CCU RESET

                    |
                    |
A ABORT ROUTINE     |
                    |
C CANCEL REQUEST    |
                    |
RX..X    REPLY:     |
 ENTER R            |
 FOLLOWED BY X..X   |
 (IFT MESSAGE),     | REQUEST: .... .                DIAG RUNNING
 THEN PRESS SEND.   | OPTIONS: S   NW C1   R1        ROUTINE ....
 (REPLY MAY BE A    | REQUESTED CCU STOP OCCURRED.
 NULL STRING)       |
                    | ENTER REQUEST ACCORDING TO THE DIAG MENU
                    | ==> R
===>                                                 START COMPLETED
```

DIAGNOSTIC MENU DESCRIPTION

DIAGNOSTIC REQUEST MENU

This screen is presented after you enter 'D' on service diskette primary menu display (for initial screen, see "How to Run Offline Diagnostics" earlier in this chapter).

```
PROCESS  STOP-CCU-CHKSERVICE MODE
          BYP-ADP-CHK


D:DIAGNOSTICS                                    SP:CCU STOP     Q:DATE/TIME
                                  U:UTILITY PGM  ST:CCU START    T:TERMINATE
E:ERROR LOG                                      RS:CCU RESET

DIAG  |ADP# |LINE |
                  |
1 ALL             |
2 CCU             |
3 IOCB            |
4 CA  |1-> 6|     |
5 TSS |1->16|0->31|
6 OLT |1-> 6|     |
7 TRSS|6->16|1->4 |
       AND        |                                  DIAG - RUN INIT
                  |
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                  | ENTER REQUEST ACCORDING TO THE DIAG MENU
                  | DIAG==>        ADP#==>     LINE==>     OPT==> N
===>
```

Selection example:

DIAG==> 5 ADP#==> 7 LINE==> 14 OPT==> Y

In this example, the TSS diagnostic group is selected (5), and tests line 14 of scanner 7. The modify option is requested.

1 **ALL**

The object of this selection is to run the diagnostics without manual intervention.

It causes the offline diagnostics to run in the mandatory sequence. The console screen shows the progress of the diagnostic by updating the DIAG STATUS AREA every time a new routine is entered. Channel adapters, scanners, and telecommunication lines are all tested in turn if present in the 3725 CDF.

The OLTs and the manual routines are not run when ALL is selected. This selection provides a quick check that the communication controller is working.

2 **CCU**

To run the CCU diagnostics.

3 **IOCB**

To run the redrive and the IOCB diagnostics.

4 **CA |1-> 6|**

You may select a channel adapter in the range given. If you do not select a channel adapter, all are tested in turn up to the last one defined in the 3725 CDF.

5 **TSS |1->16|**

When DIAG 5 is entered with no further parameters, all the lines on every scanner are tested if defined in the 3725 CDF. You may test all the lines attached to a scanner, or one line only on a particular scanner. For detailed information and line testing possibilities, see Chapter 5.

6 **OLT |1-> 6|**

This selection loads the channel adapter responder program into CCU storage, and responds to the requests of the host OLTs. You must select a channel adapter defined in the 3725/3726 CDF.

7 **TRSS|1->16|**

This selection allows you to test the TRA (TRM and TIC cards). If a TIC number (1 to 4) is selected, only the TRM and the corresponding TIC card are tested. If no TIC number is specified, all TIC cards are tested.

OPT Press the Y (yes) key to display the modify menu. The default option is N (no); it means that SELECTION MODIFY MENU is not requested.

Default options are:

- S (stop on first error)
- NW (no wait before execution of each routine)
- C1 (cycle = 1)
- R1 (execute each routine once)

RX..X A diagnostic may require you to enter parameters. The request is displayed on the message line. This requires a full reply: You **must not** reply with a null entry or 'R' only, but with 'R' and the parameters as follows:

ENTER LEVEL YOU WANT: 01, 02, 03, 04, 05

Reply R02 if you want to run the routine on CCU interrupt level 2.

Manual Intervention

A diagnostic may also require you to follow a manual intervention associated with troubleshooting (the associated procedure is provided in the MIM Part 2).

Example:

RECONNECT RDV3 THEN PRESS SEND

The jumper placed on redrive card number 3 must first be removed, then the SEND key must be pressed.

DIAGNOSTIC SELECTION MODIFY MENU

This is the screen presented after DIAG REQUEST MENU when the MOD field has been updated to the Y value, and before the selected diagnostic has been started.

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK

D:DIAGNOSTICS                                  SP:CCU STOP    Q:DATE/TIME
                              U:UTILITY PGM    ST:CCU START   T:TERMINATE
E:ERROR LOG                                    RS:CCU RESET

                    |
                    |
G GO                |
C CANCEL REQUEST    |
                    |
M MODIFY OPTIONS:   |
 S/LS/AL/ALS/B/DM   |
 NW/W               |
 C1/CNNN/C          | REQUEST: PB                      DIAG - RUN INIT
 R1/RNNN            | OPTIONS: S   NW C1   R1
                    |
                    |
                    | ENTER REQUEST ACCORDING TO THE DIAG MENU
                    | ==> M DM W
===>
```

DIAGNOSTIC EXECUTION MODIFY MENU

This is the screen presented after the ATTN key has been hit during the running of the diagnostic, and when the break has been received by the control program.

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK

D:DIAGNOSTICS                                  SP:CCU STOP    Q:DATE/TIME
                              U:UTILITY PGM    ST:CCU START   T:TERMINATE
E:ERROR LOG                                    RS:CCU RESET

                    |
                    |
A ABORT ROUTINE     |
G GO                |
C CANCEL REQUEST    |
R RERUN REQUEST     |
M MODIFY OPTIONS:   |
 S/LS/AL/ALS/B/DM   |
 NW/W               |
 C1/CNNN/C          | REQUEST: TSS 1                TSS  DIAG RUNNING
 R1/RNNN            | OPTIONS: S   NW C1   R1            ROUTINE PA04 ADP 01
                    |
                    | BREAK RECEIVED
                    | ENTER REQUEST ACCORDING TO THE DIAG MENU
                    | ==> G
===>
```

The modify options are separated by a slash (/) on each line of the menu. This means that on a given line, the options are mutually exclusive. However, you may select several options, one per line of the menu.

If you select more than one option for the same line, only the last one is accepted; if you do not select any option, the initial values are:

```
S
NW
C1
R1
```

To change options, you must enter the following:

```
M DM W
|| |  |
|| |  └------- Space required
|| |
|| └---------- Space required
|| |
|| M is required
```

To leave the modify option (M), enter any command (A, C, G, or R). The request is executed.

OPTIONS

S STOP ON FIRST ERROR

This is the default option.

The diagnostic request is executed. On detection of the first error, testing stops and the error information is displayed.

You may then use any command of the execution modify menu to:

- Abort the routine and continue the request from the next routine
- Continue the request
- Cancel the request
- Rerun the request from the beginning
- Modify the 'stop' option to another, or add any compatible option.

If you choose to continue the request by typing G (go), the request continues from the error until a second error is detected; at that time the error is displayed and the request stops again.

LS LOOP ON FIRST ERROR WITH STOP

The diagnostic request is executed until the first error is detected. The DCF then displays this error and stops. Entering G (go) causes the DCF to loop on the error, and to stop and display when the same error is detected again.

The loop is maintained on this error display whether the error which initiated the loop remains or not. If a new error appears inside the loop, it is displayed as N ERC (new ERC) and N RAC (new RAC); but even then the loop is maintained on the first error detected.

At each display stop, you may enter one of the commands of the 'execution modify' menu to:

- Abort the routine
- Continue the loop
- Cancel the request
- Rerun the request
- Modify the LS option, or add any compatible option

AL AUTOMATIC LOOP ON ERROR

The diagnostic request is executed until the first

error is detected. The DCF displays this error and starts looping automatically on the error. The loop is maintained on this error whether the error which initiated the loop remains or not. If a new error appears inside the loop, it is displayed as N ERC (new ERC) and N RAC (new RAC), but even then, the loop is maintained on the first error detected.

As there is no stop once the loop has been initiated you have no other way to regain control than to press the ATTN key. The message BREAK RECEIVED then appears. The request is stopped and you may use one of the commands of the 'execution modify' menu:

- Abort routine
- Go
- Cancel request
- Rerun request
- Modify options

  For example, you can modify the AL option to the S option, then enter G (go). The request continues to completion or until an error is detected and displayed.

ALS AUTOMATIC LOOP ON ERROR WITH NEW ERROR STOP

The diagnostic request is executed until the first error is detected. The DCF displays this error and begins to loop automatically on the error. The loop is maintained on this error display, whether the error that initiated the loop remains or not. If a new error appears inside the loop, it is displayed as N ERC (new ERC) and N RAC (new RAC); the DCF stops on this display.

You may now enter any command of the 'execution modify' menu.

For example:

- Go restarts the loop (ALS option still active)
- M S changes the ALS option to the stop on error option; entering 'go' then causes the DCF to continue the request sequentially to completion or until the next error is detected and displayed.

  If no N ERC or N RAC occurs (no new error detected during the loop), there is no other way for you to regain control other than by pressing the ATTN key. The message BREAK RECEIVED appears, the request is stopped, and you may then enter any command of the 'execution modify' menu:

  - Abort routine
  - Go
  - Cancel request
  - Rerun request
  - Modify options

    For example, you can 'modify' the ALS option to the S option, then enter G (go). The request continues until 'request complete' is displayed, or until an error is detected and displayed.

B BYPASS ERROR STOPS

The diagnostic request is executed until an error is detected. On detection of the error, the DCF displays the error information; testing then resumes automatically until another error is detected or until the request is complete. Every error detected is

displayed in this way. You have no other way to regain control before the end of the request other than to press the ATTN key.

The message BREAK RECEIVED appears, the request is stopped, and you may then enter any command of the 'execution modify' menu:

- Abort routine
- Go
- Cancel request
- Rerun request
- Modify options

DM DISPLAY MULTIPLE ERRORS

The diagnostic request is executed until an error is detected. On detection of the error, the DCF displays the error information, aborts the routine, and automatically starts the next routine. In this way, only the first error detected in each routine is displayed. Execution then continues automatically until the request is complete. You have no other way to regain control before the end of the request other than to press the ATTN key.

The message 'break received' appears, the request is stopped, and you may then enter any command of the 'execution modify' menu:

- Abort routine
- Go
- Cancel request
- Rerun request
- Modify options

NW/W

W WAIT BEFORE EXECUTION OF EACH ROUTINE

The execution of the diagnostic request stops before each routine. Entering G starts the next routine in sequence.

The message ROUTINE READY TO START is displayed and you may use any command of the 'execution modify' menu.

NW NO WAIT BEFORE EXECUTION OF EACH ROUTINE

This is the default option. It cancels the W option, and allows the request to execute without a stop before routine execution.

C1/CNNN/C

C CYCLE ON REQUEST

The DCM executes the entire request, and then automatically restarts it.

This 'cycle' on the entire request continues indefinitely unless you press the ATTN key and change the option to C1.

C1 CYCLE = 1

This is the default option.

The request is executed **once**, and ends with the REQUEST COMPLETE message.

Cnnn CYCLE nnn TIMES ($1 \leq nnn \leq 255$)

The request is executed nnn times.

After nnn cycles, the request ends and the message REQUEST COMPLETE is displayed.

R1/Rnnn

Rnnn REPEAT EACH ROUTINE nnn TIMES

Each routine is executed nnn times (1 ≤ nnn ≤ 255) before the next routine is executed.

R1 REPEAT EACH ROUTINE ONCE

This is the default option. Each routine is executed once.

If you have selected Rnnn and want to return to the default option, modify Rnnn to R1.

COMMANDS

ABORT GO CANCEL RERUN

A ABORT ROUTINE

The current routine is aborted. The next routine in sequence is entered, or the request is complete if the abort occurred on the last routine.

G GO (Selection Modify Menu)

After your request is entered on the screen, GO starts the execution.

G GO (Execution Modify Menu)

After your request is entered on the screen, GO resumes the execution.

C CANCEL

The current request is canceled, allowing a new request. The current set of options is reset.

R RERUN

The diagnostic request is restarted from the beginning. The current set of options is not changed.

DIAGNOSTIC WORK AREA DESCRIPTION

Common and detailed diagnostic information is available on the diagnostic screens.

COMMON INFORMATION

The common information area is permanently updated, depending on the monitoring of the IFT diagnostic runs. The following screen is presented after an error has been found by the diagnostic, and the selected option is stop on error (with or without the loop option). This screen is an example, and should not be used for troubleshooting.

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK

D:DIAGNOSTICS                                  SP:CCU STOP    Q:DATE/TIME
                              U:UTILITY PGM    ST:CCU START   T:TERMINATE
E:ERROR LOG                                    RS:CCU RESET

                     |
                     |******************
A ABORT ROUTINE      |* RAC  185        *
G GO                 |* ADDR 02 20      * ERR BIT  0001
C CANCEL REQUEST     |* ERC  QB010611   *                    ERROR COUNT  00001
R RERUN REQUEST      |******************
M MODIFY OPTIONS:    |
 S/LS/AL/ALS/B/DM    |
 NW/W                | START 00:10:57   STOP 00:11:06
 C1/CNNN/C           | REQUEST: TSS 1                TSS  DIAG RUNNING
 R1/RNNN             | OPTIONS: S   NW C1  R1             ROUTINE QB01 ADP 03 L 20
                     |                                                 LINE AD 052
                     | *** ERROR FOUND ***
T COMMAND BARRED     | ENTER REQUEST ACCORDING TO THE DIAG MENU
CCU FNCTN BARRED     | ==> G
===>
```

ADDR: Indicates the logical address of the failing element in the controller. The first two digits give the CA, RDV, or scanner number; The second two digits give the line address, if any. MIM Part 2 defines the ADDR field in the RAC index tables.

CE REPLY AREA: Initial request, command, answer to diagnostic message.

Command Barring: One of the following lines, or both, may appear on the screen:

- T COMMAND BARRED

  The action to be performed is displayed in the diagnostic menu.

- CCU FNCTN BARRED

DCF MESSAGE AREA: Enter request according to the diagnostic menu.

DCF RUNNING REQUEST: Initial request and current run options

```
REQUEST: TSS 1
OPTIONS: S   NW C1   R1
```

DCF RUN STATUS: Break received, error reporting, and so on, for example :

*** ERROR FOUND ***

DIAG RUN STATUS: The information has the following format:

```
XXXX DIAG YYYYYYYYY
     ROUTINE aann          ADP mm L pp
                           LINE AD qqq
```

where XXXX can be:

CCU
IOCB
CA
TSS

and YYYYYYYY can be:

- RUN INIT
(DCF initialization phase)
- CCU INIT
(CCU initialization phase)
RUNNING
RERUNNING
CANCELED
ENDED
UNXPTD.ERR
HUNG

```
ROUTINE QB01          ADP 03 L 20
        || |          |     |
        || Routine (01)|     Line address 20
        |Section (B)   |     in scanner
        IFT (Q)        Scanner
                       (or CA number)

                       LINE AD 052
                       Line address 052 in controller
                       (000 to 255)
```

ERC (Error Reference Code): This indicates whether you are working on the same fault or a new one (after a FRU replacement for example). It enables you to loop on one specific error only, disregarding all others or new ones, if any. The first four digits show the IFT number, section number, and routine number. The second four digits indicate the error number.

```
GB 02                          78F6
||  └─────> Routine            Error number
|└────────> Section
└─────────> IFT
```

ERR BIT: For the way in which to use its contents, see MIM Part 2.

DIAGNOSTIC MESSAGE AREA Message from current IFT routine (blank in this routine)

RAC (Repair Action Code): This is used with the repair action code index in the MIM Part 2. Refer to this chart to determine the list of the suspected FRUs, and the procedure to follow for their replacement. FRUs are identified by their generic names (for example, LIC3). Look up the FRU name in the FRU correspondence table in the MIM Part 2. Use the logical FRU address as required to find out the corresponding FRU location in your machine.

With the line address, and using the scanner board information tables starting on MIM-1, page 4-120, locate the different cards used by this line.

TIMING AREA:

- Indicates the initial time (starting from 00:00:00 because of MOSS maintenance IML).

Indicates the time of the every stop (for stop on error, request complete, or request canceled).

DETAILED DIAGNOSTIC INFORMATION

There are two types of display: DIAG ERROR REPORTING and DIAG/DCF UNEXPECTED ERROR. These screens are examples, and should not be used for troubleshooting.

DIAG ERROR REPORTING DISPLAY

This is the screen presented after an error has been found by the diagnostic, and the chosen option is STOP on ERROR.

Note: Fields having characters and dots (for example, LOOP Count...) are displayed when necessary. They are all shown here in order to indicate the maximum available information displayed on the screen in case of error reporting.

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK

D:DIAGNOSTICS                                    SP:CCU STOP    Q:DATE/TIME
                                U:UTILITY PGM    ST:CCU START   T:TERMINATE
E:ERROR LOG                                      RS:CCU RESET

                    |
                    |**************** EXP DATA ........      LOOP COUNT   .....
A ABORT ROUTINE     |* RAC  18A      * RCV DATA ........      LOOP ERR.CNT .....
G GO                |* ADDR 01 02    * ERR BIT  0001....
C CANCEL REQUEST    |* ERC  QA060411 * MASK     ........      ERROR COUNT  00001
R RERUN REQUEST     |**************** ADDIT INFO:             CYCLE COUNT  .....
M MODIFY OPTIONS:   | N RAC ...        .....................   REPEAT COUNT ...
 S/LS/AL/ALS/B/DM   | N ERC ........   .....................
 NW/W               | START 00:01:14   STOP 00:01:27
 C1/CNNN/C          | REQUEST: Q     1              TSS  DIAG RUNNING
 R1/RNNN            | OPTIONS: S   NW  C1  R1            ROUTINE QA06 ADP 01
                    |
                    | *** ERROR FOUND ***
                    | ENTER REQUEST ACCORDING TO THE DIAG MENU
                    | ==> A
===>
```

N ERC: means new ERC.

N RAC: means new RAC.

After a loop option is selected, a first error causes a loop to be maintained. If an error different from the first one occurs, it is displayed as N ERC and N RAC.

```
EXP DATA    0024 0000
RCV DATA    1020 A00A
ERR BIT     0004 0000
MASK        0FFF 0000
```

These four lines of information work together: any discrepancy between the EXP (expected) DATA and the RCV (received) DATA is taken into account if the corresponding MASK bit is on.

ERR BIT can also be displayed alone. In this case, EXP DATA, RCV DATA and MASK fields are not displayed. For the way in which to use its contents, see MIM Part 2.

ADDIT INFO: means additional information. The meaning varies with the routine that displays the additional information.

When the ADDIT INFO field is used by a routine, the description of its contents is found in the routine diagnostic group.

LOOP COUNT: is incremented by one prior to the execution of a routine when looping on an error. The displayed value is incremented every time the loop is entered, whether the error occurs or not. The loop count is reset at the beginning of a request, at a routine start, or at any loop option change.

LOOP ERR CNT: means loop error count. The displayed value is incremented only when the referenced error (first error) occurs in the loop. The loop error count is reset at the beginning of a request, at a routine start, or at any loop option change. Comparing LOOP COUNT and LOOP ERR CNT values helps to determine the number of intermittent error occurrences.

ERROR COUNT: indicates the count of any error encountered while a request is running. A new request resets the error count. The displayed value is updated while the diagnostic is running.

CYCLE COUNT: indicates the current count of the Cnnn option you specified. The displayed value is updated while the diagnostic is running.

REPEAT COUNT: indicates the current count of the Rnnn option you specified. Modifying the option or starting a new request resets the repeat count.

DIAGNOSTIC UNEXPECTED ERROR DISPLAY

This is the screen presented after an UNEXPECTED ERROR has been found by the diagnostic. This is a **major** error; the diagnostic run cannot continue.

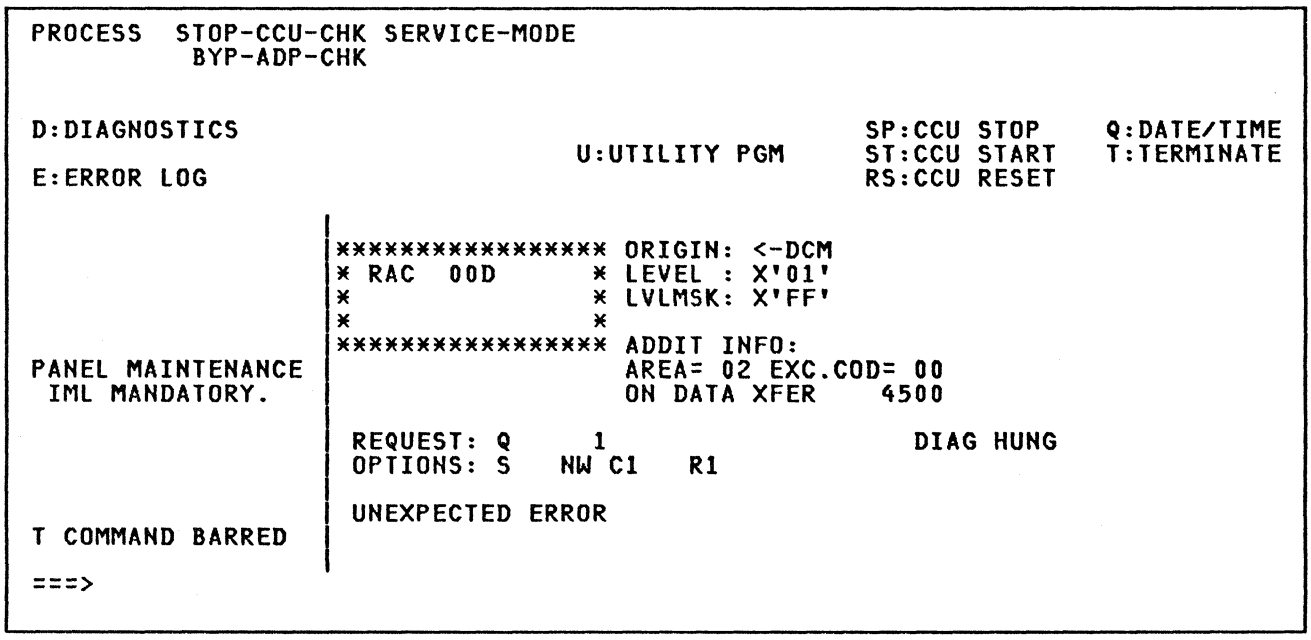
Note: In this case, NEW REQUEST is proposed to the CE. This means that the following commands are allowed: CCU FUNCTIONS and TERMINATE COMMAND (PF keys).

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK


D:DIAGNOSTICS                                        SP:CCU STOP    Q:DATE/TIME
                                    U:UTILITY PGM    ST:CCU START   T:TERMINATE
E:ERROR LOG                         X:CCU X FNCTN    RS:CCU RESET

DIAG  |ADP# |LINE |
                  |****************** ORIGIN: MOSS<-IFT
1 ALL             |* RAC  868       * LEVEL : X'01'
2 CCU             |*                * LVLMSK: X'FF'
3 IOCB            |*                * ROUTINE BH03
4 CA  |1-> 6|     |****************** ADDIT INFO:
5 TSS |1->16|0->31|                   RC= 84
6 OLT |1-> 6|
7 TRSS|6->16|1->4  | START 00:09:18 STOP 00:10:17
      AND         | REQUEST: BH03                  CCU  DIAG UNXPTD.ERR
                  | OPTIONS: S   NW C1   R1             ROUTINE BH03
OPT = Y IF MODIFY |
 OPTION REQUIRED  | UNEXPECTED ERROR
                  | ENTER REQUEST ACCORDING TO THE DIAG MENU
                  | DIAG==>       ADP#==>     LINE==>     OPT==> N
===>
```

DCF UNEXPECTED ERROR DISPLAY

This is the screen presented after an UNEXPECTED ERROR has been found by the DCF. This is a **major** error; the diagnostic run cannot continue. This screen is an example, and should not be used for troubleshooting.

Note: The displayed RAC may have multiple causes. The ADDIT INFO zone (AREA and EXC.COD) then gives more detailed information.



DESCRIPTION OF FIELDS FOR UNEXPECTED DCF RAC

In catastrophic cases, such as erroneous logical status, or return code not null after an I/O operation, the diagnostic control facility (DCF) displays a screen containing a special repair action code (RAC) referring to an **unexpected error.**

These RACs may be requested by any DCF component:

- DCM: Diagnostic Control Monitor
- CP MOSS: Command Processor - MOSS
- CP CCU: Command Processor - CCU
- CP CSP: Command Processor - CSP

In principle, an unexpected error RAC is meaningful only in perfectly debugged code. Therefore, during an investigation into an unexpected error, as a last possibility, you should suspect a **software error** in the DCF.

The RAC lists on the following pages assume that the DCF is free of software errors. The error can be caused by any of the following modules: diskette generation, diskette copy, microcode interface tape (MCIT) transfer, or any software component external to the DCF.

DCF ADDITIONAL INFORMATION

CP Additional Information Meaning

All RACs reported by each command processor (CP) are self-explanatory; there is no standard format.

DCM Additional Information Meaning

This area is structured as follows:

- The first line indicates the exercised area (AREA=) and the current operation exception code (EXC.COD=) in this area.
- The second line displays the logical DCF status at the time of the operation. It consists of a DCF message with the following format:

  ON condition = wxyz

  where:

  condition = DCF logical status
  wxyz = DCF value

Exercized

AREA Code	Area Name
00	Control code
01	Diskette
02	Panel MIOC
03	CCU IOC
04	Timer

Exception Code Area

AREA Code	Exception Code List
00	(not used)
01	Disk adapter list
02	Panel adapter list
03	Panel adapter list
04	Timer adapter list

Condition

ON condition	AREA	Meaning
ON EVENT/STATE	00	Event wx received on state yz is rejected.
ON ROUTINE	00	Routine wxyz (for example AB01) does not exist in current IFT section header
ON SST OFFSET	00	The offset wxyz in IFT Sequence table is wrong.
ON (module name) or ON (file name)	01	Module or file: empty or not found, already open or closed
ON DATA XFER	02 03	- wxyz reflects the last MIOC command sent. - wxyz reflects the last DCM message sent to CP.

DCF REPAIR ACTION CODE INDEX

DCF RAC Range

RAC Range	DCF Component
000 to 04F	• DCM
050 to 07F	(not used)
080 to 09F	• CP MOSS
0A0 to 0BF	• CP CCU
0C0 to 0DF	• CP CSP
0E0 to 0FF	(not used)

DCM REPAIR ACTION CODE INDEX

The following RACs are generated by the DCM whatever controller part is under test. Diagnostics are loaded in the MOSS, the CCU, or the scanner.

DCM Invalid Status

RAC	Function/Area in Error	Comments
001 002 003 004 005	• Event rejected by DCM nucleus • Event rejected by CP nucleus • IFT routine not found by CP (IFT section header degraded) • Invalid SST access request • IFT routine CLASS unknown	Meaning: Error in MIOC data transfer, or CP or IFT degraded code
006 007 008 009	not used	

CCU Initialization via MIOC

RAC	Function/Area in Error	Comments
00A 00B 00C 00D 00E	• Multipurpose CCU init • Init CCU with LSSD • CCU keys setting • CCU storage test pattern • CCU Address Exception - Hardstop	Meaning: Error in MIOC data transfer
00F	not used	

Load Module Loading from Diskette to MOSS

RAC	Function/Area in Error	Comments
010 011 012 013 014 015 016 017 018 019 01A 01B 01C	• Overlay • IFT Section - on OPEN - on CLOSE - on READ - on LOAD • Data FILE for IFT - on OPEN - on CLOSE - on READ - on LOAD • Command Processor 'CP' - on OPEN - on CLOSE - on READ - on LOAD	Meaning: Disk error, or Load module not found or Load module degraded

File Reading from Diskette

RAC	Function/Area in Error	Comments
 01D 01E 01F	• CDF - on OPEN - on CLOSE - on READ	Meaning: Disk error, or CDF file not found, or CDF file degraded

The following RACs are generated by the DCM when the command processor (CP) and the diagnostic IFTs (IOCB or CA IFTs) are loaded in the CCU.

Loading Data from MOSS to CCU via MIOC

RAC	Function/Area in Error	Comments
020 021 022 023 024 025	• CDF • SST • Replied data (for IFT) • CP CCU • IFT section • IFT data file	Meaning: Error in MIOC data transfer

Sending Mailbox In from MOSS to CCU via MIOC

RAC	Function/Area in Error	Comments
026	• MIOC error on sending DCM message (exception code not 00) or • Timeout on CP CCU acknowledge (exception code 00)	Meaning: Error in MIOC data transfer, or Erroneous LOOP in IFT CCU, or DCF degraded code
027	not used	

Receiving Mailbox Out from CCU to MOSS via MIOC

RAC	Function/Area in Error	Comments
028	• MIOC error in getting CP CCU answer (exception code not 00)	Meaning: Error in MIOC data transfer

Initialization from MOSS via MIOC

RAC	Function/Area in Error	Comments
029 029 02A	• CCU, IOCB IFT initialization • CP CCU initialization	Meaning: Error in MIOC data transfer
02B to 02F	not used	

The following RACs are generated by the DCM when the command processor (CP) and the TSS IFTs are loaded in the scanner.

Loading Data from MOSS to CCU via MIOC

RAC	Function/Area in Error	Comments
030 031 032 033 034 035	• CDF • SST • Replied data (for IFT) • IFT section • IFT data file • DCM message and data for CP	Meaning: Error in MIOC data transfer
036 037 038	not used	

Loading CP CSP from CCU to CSP

RAC	Function/Area in Error	Comments
039 03A	• MIOC CP CSP loading command failed (exception code not 00) • Cycle steal AIO Write error (CSP ROS trying to load CP in CSP)	Meaning: Error in MIOC data transfer, or Error in CCU-IOC, or Error on IOC-BUS
03B to 04A	not used	
04B 04C 04D	• Wrong section header in TSS • Hardcheck in CCU • DCM dispatched for nothing	Meaning: IFT degraded microcode - DCM/MOSS degraded microcode

Timeout on Communications with CP via IOC-BUS

RAC	Function/Area in Error	Comments
04E	• NO CP CSP acknowledge on DCM message or data (in ADDIT INFO, data transfer field gives last DCM msg)	Meaning: Erroneous loop in IFT CSP, or CP or IFT code degraded
04F	• Halt condition appears on IOC BUS, during CP CSP process. (in ADDIT INFO, data transfer field gives last DCM msg)	Meaning: After halt condition, communications are not terminated correctly

COMMAND PROCESSOR RAC INDEX

The following RACs are generated by the command processor (CP) when the CP and the CCU diagnostics are loaded and run in the MOSS.

CP Invalid Status (MOSS)

RAC	Function/Area in Error	Comments
080	not used	
081 082	• IFT domain active during CP process • CP domain active with no CP task requested	Meaning: CP degraded code

Unexpected Interrupts (MOSS)

RAC	Function/Area in Error	Comments
083 084 085 086 087 088 089	• No IFT int. handler defined: - at level 0 - at level 1 - at level 4 • When CP domain active: - at level 0 - at level 1 - at level 4 • Out of IFT Diags scope: - at level 0	Meaning: IFT degraded code
08A to 09F	not used	

The following RACs are generated by the CP when the CP and the diagnostics (IOCB or CA IFTs) are loaded and run in the CCU.

Unexpected Interrupts (CCU)

RAC	Function/Area in Error	Comments
0A0	• Unexpected interrupts at levels 1 or 4.	Meaning: IFT degraded code
0A1 to 0BF	not used	

The following RACs are generated by the command processor (CP) when the CP and TSS diagnostics are loaded and run in the scanner.

CP/Macros/IFT (Scanner)

RAC	Function/Area in Error	Comments
0C0	• CP/IFT communication lost (levels 0, 1, 2 or 7)	Meaning: CP or IFT degraded code

CP/DCM via IOC Bus (Scanner)

RAC	Function/Area in Error	Comments
0C1	• Invalid logical or physical status on IOC-BUS. (defined in second line of ADDIT INFO. area) or • Halt condition appears on IOC-BUS, but communications have been terminated correctly (Halt appears in 2nd line of ADDIT INFO area)	Meaning: Error on IOC BUS, or Error in CCU IOC, or CP degraded code
0C2	not used	

CP Invalid Status (Scanner)

RAC	Function/Area in Error	Comments
0C3 0C4 0C5	• CP wrongly requested interrupt level 0. • IFT domain active during CP process (level 7). • CP domain active with no CP task requested (level 7).	Meaning: CP degraded code

Unexpected Interrupts (Scanner)

RAC	Function/Area in Error	Comments
0C6 0C7 0C8 0C9 0CA 0CB 0CC 0CD 0CE	• No IFT int. handler defined: - at level 0 - at level 1 - at level 2 • When CP domain active: - at level 0 - at level 1 - at level 2 - at level 3 - at level 4 - at level 5	Meaning: IFT degraded code
0CF to 0DF	not used	

CHAPTER 2. CCU DIAGNOSTICS

Table of Contents for Chapter 2

CHAPTER 2. CCU DIAGNOSTICS

The CCU diagnostic group is divided into the following internal function tests (IFTs) that test:

- Direct operations (IFT A)
- Indirect operations (IFT B)
- Basic instruction set (IFT C)
- Full instruction set (IFT D)
- Full storage (IFT E)
- Branch trace/address compare (IFT F)

REQUIREMENTS

The MOSS must be IMLed and running before testing the CCU.

When under test, the CCU is dedicated to diagnostic mode.

To get meaningful error information, the CCU IFTs must be run in sequence.

Because the CCU diagnostics modify the LSSD strings, the CCU services must not be used while testing the CCU (for details of LSSD registers, see Chapter 14 in MIM Part 1). **Warning: Make sure that all CA switches are set to DSBL (disabled).**

SELECTION

For running offline diagnostics, see Chapter 1 of this manual.

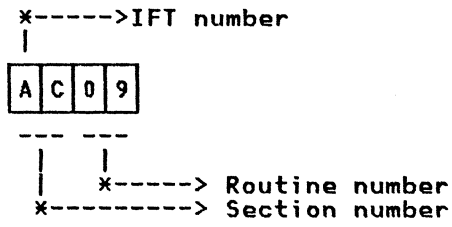
DIAG==>_

2	CCU group Selected
X	Specific IFT X in this group
XY	Specific section Y in IFT X
XYZZ	Specific routine ZZ in Section XY

For specific section and routine selection, see routine lists on the following pages and in Chapter 1.

Move the cursor from its initial position (DIAG==>) to the next after each parameter is entered. To skip a parameter entry, press the --> key.

The routine number contains the IFT number and the section number as follows:



ADP#==>_ (not applicable)

LINE==>_ (not applicable)

OPT==> N For option display and description, see Chapter 1.

The following restrictions apply when running the CCU diagnostic group.

The modify options 'cancel', 'abort', and 'rerun' do not always reset error information in the CCU diagnostics. To eliminate this misleading error information, press the Power-On switch to force a reset and a MOSS re-IML. The Power Control switch must be set to local for the reset to occur.

Note: The results of running a selected section or routine are valid only if the preceding IFTs, sections, and routines of the diagnostic group run error-free.

ADDITIONAL FIELD DESCRIPTIONS

Field	Description	Page
CMSA or A	CCU-to-MOSS Status A Reg	14-040
CMSB or B	CCU-to-MOSS Status B Reg	14-040
CMSC or C	CCU-to-MOSS Status C Reg	14-040
MCC S1	MCC Status 1 Reg	14-030
MCC S2	MCC Status 2 Reg	14-030
RC	Return code if MOSS operation is rejected (see next table)	
PCW ADDR	Processor Control Word Address	

RC Bit	Meaning
0	Not OK return from CAC (always 1)
1	Adapter down
2	LSSD error (residual count > 7)
3	String select error
4	Invalid register selection
5	Busy bit is on
6	IOC error
7	Abort

On the following screen, section AH is selected without option modification.

Press SEND to execute the request.

Read what the DCM displays in the work area and proceed with the next action according to the displayed menu or message.

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK


D:DIAGNOSTICS                                    SP:CCU STOP    Q:DATE/TIME
                                  U:UTILITY PGM  ST:CCU START   T:TERMINATE
E:ERROR LOG                                      RS:CCU RESET

DIAG  |ADP# |LINE |
      |     |     |
1 ALL |     |     |
2 CCU |     |     |
3 IOCB|     |     |
4 CA  |1-> 6|     |
5 TSS |1->16|0->31|
6 OLT |1-> 6|     |
7 TRSS|6->16|1->4 |
       AND    |                                  DIAG - RUN INIT
      |     |     |
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> AH    ADP#==>     LINE==>     OPT==> N
===>
```

CCU DIAGNOSTIC GROUP RUNNING TIME

When the diagnostic request is 2, the total running time is 18 minutes 50 seconds.

MANUAL INTERVENTION ROUTINES

- **Routine AB02** is used when scoping between the MOSS and the CCU.
- **Routine AB03** is used to provide a FRU List for a CCU hardcheck failure.
- **Routine BI04** is used when scoping the CCU storage.
- **Routine BI05** is used to test the remote power off capability.

Routines AB02, AB03, BI04, and BI05 will not run unless specifically selected.

OPTION SELECTION RESTRICTION

Loop on error is not allowed when an error occurs in routine BK04.

Manually-Invoked Routines

- **Routine AJ01** is used when the standard CCU diagnostic group does not isolate the failure. See in the MIM Part 2, Chapter 3, the CCU extended replacement procedure.
- **Routine BO01** is used to analyse the CCU storage.

Routines AJ01 and BO01 will not run unless specifically selected.

MANUAL SELECTION RESTRICTIONS

Routines DA61, DB67, DC6F and DD76 must not be selected. These routines check that the general purpose register contents set by routines DA01, DB02, DC03, and DD04 have not changed after the CCU instruction set is tested.

Routine BJ03 must not be selected; it uses information from BJ02.

DIRECT OPERATION - IFT A

This IFT is loaded in the MOSS storage and tests the MIOC card using direct operations. The following are tested:

- MCC/MIOC card connection
- LSSD mechanism (ending with the CCU LSSD initialization)
- High/low level interrupt request functions
- MIOC parity circuits
- Registers that can be read/written with direct operations
- Hardcheck error path
- Interrupt disable functions
- CCU hardstop bypass function
- MDOR (on DFL1-x cards)
- CCU-to-MOSS status registers E and F

RUNNING TIME

Run Init	10 sec
AA	9 sec
AB	3 sec
AC	11 sec
AD	7 sec
AE	15 sec
AF	7 sec
AG	9 sec
AH	10 sec
AI	18 sec
AJ	Manually invoked
AK	1 min 46 sec
Total	3 min 25 sec

MANUAL INTERVENTION ROUTINES

AB02: MOSS-to-CCU scoping

AB03: CCU BERs analysis procedure

Manually-Invoked Routine

AJ01: CCU SRL test with initial values

MESSAGES

None

INDIRECT OPERATION - IFT B

This IFT is loaded in the MOSS storage and tests the MIOC card using operations. The following are tested:

- Branch trace/address compare registers
- Local stores
- CCU data flow registers
- Wrap branch trace mechanism
- Input and output X'7X' instructions
- Storage scanning
- ECC mechanism
- CCU main storage
- Remote power off

RUNNING TIME

Run Init	
BA	10 sec
BB	1 min 15 sec
BC	18 sec
BD	5 sec
BE	7 sec
BF	5 sec
BG	6 sec
BH	3 min 58 sec
BI	3 sec
BJ	3 sec
BK	33 sec
BL	1 min 14 sec
BM	14 sec
BN	10 sec
B001	6 sec
	manually invoked
Total	8 min 27 sec

MANUAL INTERVENTION ROUTINES

BI04: CCU/Storage Scoping Routine

BI05: Network Power Off Test

Manually-Invoked Routine

B001: Storage Solid 1-Bit Error Detection

MESSAGES

None

BASIC INSTRUCTION - IFT C

This IFT is loaded into the MOSS storage together with an IFT responder in the CCU storage. It verifies the basic instruction set step-by-step.

RUNNING TIME

Run Init	56 sec
CA	8 sec
CB	11 sec
CC	8 sec
CD	8 sec
CE	8 sec
CF	7 sec
CG	9 sec
CH	8 sec
Total	2 min 3 sec

MESSAGES

None

FULL INSTRUCTION - IFT D

This IFT is loaded into the MOSS storage together with an IFT responder in the CCU storage.

It exercises the full instruction set at every interrupt level. It checks the general registers and the changing of the interrupt level.

This IFT does not exercise IOH and IOHI instructions.

- The 'IOHI' instruction is tested with the IOC bus IFTs.
- The 'IOH' instruction is tested with the RDV IFTs.

Note: The 'Exit' instruction is tested with interrupt level swapping in the routines of the DA80 series of this IFT.

SPECIAL SEQUENCE OF IFT D ROUTINES

The routines of this IFT are listed in the sequence of the routine number, not taking into account the section. For example, DA80 is after DD78.

In the routine numbers, DX means DA, DB, DC, DD, or DE. (For example, DX10 refers to DA10, DB10, DC10, DD10, or DE10.

RUNNING TIME

Run Init	10 sec
DA	51 sec
DB	47 sec
DC	48 sec
DD	48 sec
DE	44 sec
Total	4 min 8 sec

MESSAGES

None

FULL STORAGE - IFT E

This IFT is loaded into the CCU storage and responds to the diagnostic control monitor (DCM) requests through the communication processor (CP) in the MOSS.

This IFT tests the storage extensively.

Running Time

Run Init EA EB	10 sec 17 sec 4 sec
Total	31 sec

Messages

None

BRANCH TRACE/ADDRESS COMPARE - IFT F

This IFT is loaded into the CCU storage and responds to the diagnostic control monitor (DCM) requests through the communication processor (CP) in the MOSS.

This IFT tests the storage extensively.

RUNNING TIME

Run Init FA	10 sec 4 sec
Total	14 sec

MESSAGES

None

ROUTINES DESCRIPTION

AA01 - MOSS INOPERATIVE TEST

When the MOSS inoperative bit is set in the MCC Status Register 1, the MIOC/MCC card is disabled. Any read operation causes a level 0 interrupt.

1. To perform the test, interrupts to the MOSS are disabled, and MOSS inoperative is set in the MCC Status Register 1.

2. Read from address 0 and check the expected interrupt level 0.

ERC	Function	Error Description	RAC	Comments
0700	Verify interrupt level 0	No interrupt occurred and no bit is set in MCC Status Register 2.	802	
0701	Analyse MCC Status Register	CCU parity check bit on	803	
0702	"	MIOC/MOSS parity bit on	803	
0703	"	CCU parity check and MIOC/ MOSS parity bits on	805	
0705	"	CCU/MOSS timeout and CCU parity check bits on	802	
0706	"	CCU/MOSS timeout and MIOC/MOSS parity bits on	802	
0707	"	CCU/MOSS timeout and MIOC/MOSS parity and CCU parity check bits on	805	

AA02 - CCU PARITY CHECK LINE TEST (PART 1)

This routine checks that the setting of the Interrupt Test bit in MCC Status Register 2 raises a CCU parity check bit in MCC Status Register 2 when a read operation is performed. The Interrupt Test bit forces bad parity on the MIOC Address Bus.

1. Set Interrupt Test bit in MCC Status Register 2.
2. Read from address 0 and check the interrupt level 0.
3. Verify other possible events in MCC Status Register 2 that could cause an interrupt level 0.

ERC	Function	Error Description	RAC	Comments
0700	Test interrupt level 0	No interrupt level 0	802	
	Read MCC Status Register 2 for level 0 interrupt due to: - CCU/MOSS timeout - MIOC/MOSS parity check - CCU parity check			EXP DATA and RCV DATA fields contain respectively expected value and received value of MCC Status Register
0701		CCU parity check	802	
0702		MIOC/MOSS parity check	804	
0703		MIOC/MOSS parity check and CCU parity check	803	
0706		CCU/MOSS timeout and MIOC/MOSS parity check	840	

AA03 - CCU PARITY CHECK LINE TEST (PART 2)

This routine checks that setting the Interrupt Test bit in the MCC Status Register 2 raises a CCU Parity Check bit in MCC Status Register 2 when a write operation is performed. The Interrupt Test bit forces bad parity on the MIOC Address Bus.

1. Set the Interrupt Test bit in the MCC Status Register 2.
2. Write to address 0 and check the interrupt level 0.
3. Verify other possible events in MCC Status Register 2 that could cause an interrupt level 0.

ERC	Function	Error Description	RAC	Comments
0700	Test interrupt level 0 Read MCC Status Register 2 for level 0 interrupt due to: - CCU/MOSS timeout - MIOC/MOSS parity check - CCU parity check	No interrupt level 0	802	EXP DATA and RCV DATA fields contain respectively expected value and received value of MCC Status Register
0701		CCU parity check	803	
0702		MIOC/MOSS parity check		
0703		MIOC/MOSS parity check and CCU parity check	805	
0706		CCU/MOSS timeout and MIOC/MOSS parity check	802	
0707		CCU/MOSS timeout, MIOC/MOSS parity check and CCU parity check	805	

AA04 - MCC/MIOC TEST (PART 1)

This routine checks the parity checker and part of the address/data bus.

1. Read direct operation from address 0 and check the data which must be equal to 0 (no active bits on bus).

2. Any address/data bus parity error causes a level 0 interrupt.

ERC	Function	Error Description	RAC	Comments
0700	Read data from address 0	Data not equal to 0	811	
	Interrupt level 0 occurred		803	ADDIT INFO field displays the contents of MCC Status register 2
0701		CCU parity check		
0702		MIOC/MOSS parity check	803	
0704		CCU/MOSS timeout	806	

AA05 - MCC/MIOC TEST (PART 2)

This routine checks the parity checker and part of the address/data bus by writing (to address 0) a series of data patterns of floating zeros and ones and a pattern to exercise all the gates in the parity checker.

ERC	Function	Error Description	RAC	Comments
	Check for level 0 interrupt			ADDIT INFO field displays the contents of MCC Status register
0701		CCU parity check	803	
0702		MIOC/MOSS parity check	803	
0704		CCU/MOSS timeout	806	
0705		CCU/MOSS timeout and CCU parity check	802	

AA06 - SCAN REGISTER TEST

This routine checks the data path between the MCC and the MIOC using the Scan register as a data buffer. Test that the Scan register can be written and read with a data pattern of floating ones and floating zeros.

ERC	Function	Error Description	RAC	Comments
0700	Write/read Scan register Check for level 0 interrupt	Read data not equal to expected data.	841	EXP DATA and RCV DATA fields contains respectively written and read data
0701		CCU parity check	803	ADDIT INFO field displays the
0702		MIOC/MOSS parity check	803	contents of MCC Status register

AA07 - STRING SELECT REGISTER TEST

This routine checks that the String Select register can be read and that the data set at Power On Reset is X'A0'. Test that the String Address decode error bit (bit 1 of the String Select register) can be set and reset. Test that string address bits 4 through 7 can be set and reset without causing a string address decode error.

1. Check if the machine has just been powered on by reading the location in RAM set by the ROS IPL code and if so, verify that the value is X'A0'.

2. Write and read floating ones and zeros patterns.

ERC	Function	Error Description	RAC	Comments
0700	Read String Select register	Read data is not X'A0'	811	
0701	Check for level 0 interrupt	CCU parity check	803	ADDIT INFO field displays the contents
0702	Write/read String Select register with different patterns	Read data not equal to expected data.	811	EXP DATA and RCV DATA fields contains respectively written and read data

AA08 - STEP REGISTER TEST (PART 1)

This routine checks that the String Select register can be set and read with data X'A0'. Test that all bits in the Step register can be set and reset using address X'2F' in the diagnostic mode, and the C-Clock Stop bit set in the String Select register.

1. Write X'A0', read it back, and verify that written and read patterns are the same.
2. Set the C-Clock Stop bit and the Diagnostic Mode bit in the String Select register to prevent single-step operation.
3. Write/read floating ones and zeros patterns.

ERC	Function	Error Description	RAC	Comments
0700	Write/read the Step register with value X'A0'	Mismatch in written/read data	811	
0701	Check for level 0 interrupt	CCU parity check	803	ADDIT INFO field displays the contents of MCC Status register
0702	Write/read the String Select register with different patterns	Read data not equal to expected data.	811	EXP DATA and RCV DATA fields contains respectively written and read data

AA09 - STEP REGISTER TEST (PART 2)

This routine checks that the Diagnostic Mode bit can be turned off in the String Select register. Test that bit 7 on in the Step register starts the shift and that bit 7 off stops the shift.

1. Set the C-Clock Stop and Not Diagnostic Mode bits in the String Select register.
2. Write the Step register with different patterns.
3. Stop the clock and verify the shift.

ERC	Function	Error Description	RAC	Comments
0700	Write/read the Step register	Read data not equal to expected data.	811	EXP DATA and RCV DATA fields contain respectively expected shifted data and received shifted data

AA10 - STRING ADDRESS DECODER TEST

This routine checks that the String Address Decode error bit 1 of the String Select register detector has detected either:

- Two gated A clock enable at the same time.
- No gated A clock enable.

1. Write the String Select register using C-Clock Stop, Not MIOC Diagnostic Mode and string address 0.
2. Write the Step register with data X'01' to activate the shift mode.
3. Read the String Select register and check for an address decode error.
4. Increment the string address and loop for all 16 strings.

ERC	Function	Error Description	RAC	Comments
	Read String Select register		811	
0700		One string in error.	821 822	
0701		All but one string in error	823 824	
0702		Some strings in error.	825 826 827 830	

AA11 - STEP REGISTER TEST (PART 3)

This routine checks that the shift values put in the Step register cause the Scan register to be shifted the correct number of steps.

- Test Scan register shift mechanism.
- Test that string address X'F' shifts in ones and string address X'0' shifts in zeros.

1. Write the data in the Scan register.
2. Initialize the Step register.
3. Verify the data in the Scan register, step-by-step.
4. Test with string address X'0' and X'F'.

ERC	Function	Error Description	RAC	Comments
0700	Write/read Scan register for string address '0'	Read data not equal to expected data.	811	EXP DATA and RCV DATA fields contain respectively expected shift data and received shift data
0701	Write/read Scan register for string address 'F'			

AA12 - STEP REGISTER TEST (PART 4)

This routine checks that the address decode of X'20' through X'27' causes the correct number of shifts in the Scan register. This also checks the setting of the correct value in the Step register.

A test of the address selection mechanism is done by the LSSD operation in order to perform the scan-in. The initial data in the Scan register is X'80'. The routine changes the byte and bit counts in the Scan-In buffer in order to perform the required number of shifts. The data, after the scan-in, depends on the number of shifts.

ERC	Function	Error Description	RAC	Comments
0700	Read data from Scan register	Read data not equal to expected data.	811	EXP DATA and RCV DATA fields contains respectively expected shift data and received shift data
0701	Check for level 0 interrupt	CCU parity check	803	ADDIT INFO field displays the contents of MCC Status Register

AA13 - SCAN REGISTER WRAP TEST

This routine checks that data is correctly shifted around in the Scan register.

1. Select string address X'D' to wrap the Scan register.
2. Use a series of floating ones and zeros in the Scan register.
3. Shift the Scan register 8 times.
4. Verify that the Scan register contains the original value.

ERC	Function	Error Description	RAC	Comments
0700	Write, shift, and read back the Scan register.	Read data not equal to expected data.	811	EXP DATA and RCV DATA fields contain respectively original value and received value

AA14 - CCU LSSD STRING TEST

This routine checks all the latches of the 12 operational strings (addresses X'1' through X'C'), linked to the LSSD mechanism.

1. Select each string in turn, one at a time.
2. Write it with the following test pattern X'00FF03F03E0F0E32'.
3. Read the string.
4. Compare written and read data.

ERC	Function	Error Description	RAC	Comments
	Write/read the 12 LSSD strings with a test pattern		811 830 831	see note below
0710		One string only with data error	832 833 834	
0720		Errors found on strings 5 or 6 (DFL1-3) and on strings 7 or 8 (DFL1-2) and on strings 9 or A (DFL1-1)	835 836 837 841	
0721		Errors found on strings 3 (CTL1), 4 (CTL2), B (DFL4), and C(DFL5)		
0722		Errors found on strings 1 (MIOC), 2 (BTAC), and B (DFL4)		
0730		All strings fail		
0740		String address failure		
0750		Mixture of erroneous strings other than previous cases		

Note: The ADDIT INFO field displays the count of errors found for each string. The first line of the ADDIT INFO field gives error counts for strings X'1' through X'8', the second line for strings X'9' through X'C'.

ADDIT INFO Byte	String Address	CCU Card ID
1st line byte 0	1	MIOC
1st line byte 1	2	BTAC
1st line byte 2	3	CTL1
1st line byte 3	4	CTL2
1st line byte 4	5	DFL1-3
1st line byte 5	6	DFL1-3
1st line byte 6	7	DFL1-2
1st line byte 7	8	DFL1-2
2nd line byte 0	9	DFL1-1
2nd line byte 1	A	DFL1-1
2nd line byte 2	B	DFL4
2nd line byte 3	C	DFL5
2nd line bytes 4 to 7	N/A	0

AB01 - CCU CLOCK DISTRIBUTION TEST

This routine checks that the C-Clock is distributed to every chip of all CCU cards.

1. Write at least one bit in each chip via LSSD operations.
2. Do a single clock step.
3. Read back the latches via an LSSD operation.

ERC	Function	Error Description	RAC	Comments
0700	Verify value of selected latches in different chips on every card.	Read value not equal to expected value	81B 841 853 854 855 856 857 858 859 87A	

AB02 - MOSS/CCU SCOPING (MANUAL INTERVENTION ROUTINE)

This routine loops on given command, address, and data patterns to allow the CE to scope the MOSS/CCU interconnection.

In response to prompting messages, the CE selects one of the MOSS/CCU lines and the value to be put on the address or data bus.

ERC	Function	Error Description	RAC	Comments
		Prompting messages for CE (see MIM 1, Chapter 14, Section 2).	89F	

AB03 - CCU BER ANALYSIS PROCEDURE (MANUAL INTERVENTION ROUTINE)

This manual intervention routine provides a FRU list for a CCU hardcheck failure. It analyses the contents of registers X'7D', X'7E', and X'76' given by the BER. The contents of the register are entered by the CE on the operator console. The FRU list is then displayed.

Notes:

1. In the case of a storage 2-bit error, RAC 7CF is displayed instead of a FRU list.
2. If the FRU list is flagged, try to correlate it with other BERs (IOC bus, TSS, channel adapter) if any.

```
Message 1: 'CCU BER ANALYSIS PROCEDURE  => PRESS SEND'

Message 2: 'REPLY X7D, X7E, X76 REG VALUES (RXXXXYYYYZZZZ) =>  PRESS SEND'
  The CE should reply with 'Rxxxxyyyyzzzz', where:
  xxxx = contents of X'7D', yyyy = contents of X'7E', zzzz = contents of X'76'
   and press SEND.

Message 3: 'X7D=xxxx X7E=yyyy X76=zzzz => PRESS SEND'
```

ERC	Function	Error	RAC
0700	Storage 2-bit error: For all other checkers, a FRU list is directly given on the screen with the following analysis: -Reset CCU and IOC1 summary bits in X'7E'. -If bits on in X'76' byte 1 (bits meaningless) reset all other checkers. -If 'Z Register Parity Error', and not alone, reset it. -If 'Adapter Initiated Op', the FRU list is flagged with two asterisks. -If 'Adapter Initiated Op' and 'MOSS Initiated Op.' not alone, reset them. -If 'IOC Timeout' or 'IOC1 Bus In', reset X'76', byte 0, bits 0 through 3. -If 'Address Fetch' or 'Instruction Fetch', reset all other checkers. -If 'Invalid Op Code', reset all other checkers. -If 'ROS Parity Error', reset all other checkers: One checker: FRU list depends on this checker. More than one checker: FRU list is computed. The following message is now displayed: 'XXXX XXXX XXXX XXXX XXXX XXXX => PRESS SEND' Where XXXX is a FRU, or blank. Press SEND to end the routine.	Storage 2-bit error on	7CF

AC01 - CCU-TO-MOSS STATUS C REGISTER TEST

This routine checks that all the latches of the CCU-to-MOSS Status C register can be set and reset from the MOSS.

1. Read the register after CCU initialization.
2. Write the register with X'FF' via the LSSD and check.
3. Use floating ones and zeros to verify the setting and resetting of all latches.

ERC	Function	Error Description	RAC	Comments
0700	Test the register after CCU initialization.	A latch remained set after CCU initialization.	811 80A	
0701	Write register with X'FF'.	Invalid data using read	80A	
0702	Set latches and verify setting	Latch not set	80A	EXP DATA and RCV DATA fields contain respectively expected and received values
0703	Reset latches and verify resetting	Latch not reset	80A	

AC02 - CCU-TO-MOSS STATUS A REGISTER TEST

This routine checks that all the latches of the CCU-to-MOSS Status A register can be set and reset from the MOSS.

1. Read the register after CCU initialization.
2. Use floating ones and zeros to verify the setting and resetting of all latches.

ERC	Function	Error Description	RAC	Comments
0700	Test register after CCU initialization	A latch remained set after CCU initialization.	811 809	
0701	set latches and verify setting	Latch not set	809	EXP DATA and RCV DATA fields contains respectively expected and received status
0702	Reset latches and verify resetting	Latch not reset	809	

AC03 - CCU-TO-MOSS STATUS B REGISTER TEST

This routine checks that all the latches of the CCU-to-MOSS Status B register can be set and reset from the MOSS.

1. Read the register after CCU initialization.
2. Use floating ones and zeros to verify the setting and resetting of all latches.

ERC	Function	Error Description	RAC	Comments
0700	Test register after CCU initialization	A latch remained set after CCU initialization.	808 811	
0701	Set latches and verify setting	Latch not set	808	EXP DATA and RCV DATA fields contain respectively expected and received values
0702	Reset latches and verify resetting	Latch not reset	808	

AC04 - LOW LEVEL INTERRUPT LINE FROM CCU TO MOSS

This routine checks that the low level interrupt line from the MIOC card to the MCC card is not permanently on.

1. Disable the HLIR and LLIR using bit 1 of MCC Status Register 1.
2. Verify that LLIR bit (bit 7) in the MCC Status Register 1 is not on.
3. If bit 7 is on, read the CCU-to-MOSS Status B and C registers.

ERC	Function	Error Description	RAC	Comments
0700	Read CCU-to-MOSS Status B and C registers	Both registers contain X'00'	840	
0701	Read CCU-to-MOSS Status B and C registers	One of the registers does not contain X'00'.	840	ERR BIT field (halfword) contains CCU-to-MOSS Status B reg (byte 0) and C register (byte 1)

AC05 - LOW LEVEL INTERRUPT TO MOSS INTERCONNECTION TEST

This routine checks that each bit of CCU-to-MOSS Status Registers B and C cause the LLIR bit to be set in the MCC Status register.

1. Set one by one the latches of CCU-to-MOSS Status Register B and C using LSSD operations.
2. Verify bit 7 of MCC Status Register 1 (LLIR).

ERC	Function	Error Description	RAC	Comments
	Set the latches one-by-one in the CCU-to-MOSS Status B and C regs and verify the setting of the LLIR.			
0700		All latches failed to set the LLIR.	840	
0701		All Latches of CCU-to-MOSS Status B register failed to set the LLIR.	840	
0702		All latches of CCU-to-MOSS Status C register failed to set the LLIR.	840	
0703		Some latches of CCU-to-MOSS Status Register B and C failed to set the LLIR.	840	

AC06 - MIOC ERROR CHECK AFTER CCU INITIALIZATION

This routine checks for parity errors on the MIOC registers with the parity checker after CCU initialization.

1. Initialize the CCU and read CCU-to-MOSS Status A register to enable the gating to the MIOC Error latch.
2. Read the LSSD MIOC string.
3. Check that the MIOC Error latch and the MDOR parity error latch are both off.

ERC	Function	Error Description	RAC	Comments
	Read the MIOC Error and the			
0700	MDOR Parity Error latches.	MIOC Error latch is set	811	
0701		MDOR Parity Error latch is set	811	
0702		Both latches are set	811	

AC07 - HIGH LEVEL INTERRUPT LINE FROM CCU TO MOSS

This routine checks that the high level interrupt line is not permanently on.

1. Disable HLIR and LLIR requests using MCC Status Register 1.
2. Check that MCC Status Register 1 bit 5 is not on.
3. If the bit is on, read the CCU-to-MOSS Status A register.

ERC	Function	Error Description	RAC	Comments
0700	Read CCU-to-MOSS Status A register	Hardstop bit on in CCU-to-MOSS Status A register	840	
0701	Read CCU-to-MOSS Status A register	Other bits on in CCU-to-MOSS Status A register	843	RCV DATA halfword field contains CCU-to-MOSS Status A register (byte 0 and MCC Status Register 1 (byte 1)

AC08 - HIGH LEVEL INTERRUPT TO MOSS/CCU TEST

This routine checks that the latches of the CCU-to-MOSS Status A register (MOSS Operation Error, Address Exception, Out X'70', CA Request IPL, PGM Request IPL), an MIOC Error, and an MDOR Parity Error cause the HLIR bit to be set in MCC Status Register 2.

1. Using LSSD operations, turn on all the latches one by one, and verify that the HLIR bit (bit 5) has been set in MCC Status Register 2.

ERC	Function	Error Description	RAC	Comments
0700	Test HLIR bit in MCC Status Register 2	All latches failed to set the HLIR bit in MCC Status Register 2	840	
0701		MIOC Error and MDOR parity error latches failed to set the HLIR bit in MCC Status Register 2	809	
0702		Latches of the CCU-to-MOSS Status register failed to set the HLIR bit in MCC Status Register 2.	811	

AC09 - MOSS IOCS ERROR PATH TEST

This routine checks that the MIOH error latch sets bit 0 in the CCU-to-MOSS Status A register, which is then propagated to the MCC Status register (HLIR: bit 5).

1. Disable the HLIR and the LLIR.
2. Set the MIOH latch together with one of the following latches (of the DFL5 string) via an LSSD operation: CCU Error, Bus In Error, Storage Protect, Address Exception, and Timeout.
3. Verify the CCU-to-MOSS Status A register and MCC Status Register 1.
4. Repeat the operations, using each time a new DFL5 string latch.

ERC	Function	Error Description	RAC	Comments
	Set error bits and verify the contents of CCU-to-MOSS Status A register.			
0700		Bit 0 of CCU-to-MOSS Status A register is never set	825	
0701		Bit 0 of CCU-to-MOSS Status A register is set but the HLIR bit in MCC Status Register 1 is not set	811	
0702		Bit 0 of CCU-to-MOSS Status A register is sometimes set.	80B	

AC10 - BUS INTERCONNECTION CHECK STOP LATCH TEST

This routine checks that all the IOC error latches set the CCU Hard Check latch in the CCU-to-MOSS Status A register when the Bus Interconnection Check Stop latch is on and does not set it when it is off.

1. Turn on the IOC1 and IOC2 error latches and verify the CCU-to-MOSS Status A register.
2. Turn on the Bus Interconnection Check Stop latch in the Diagnostic Mode Control register and verify the CCU-to-MOSS Status A register.

ERC	Function	Error Description	RAC	Comments
	Read CCU-to-MOSS Status A register			
0700		Hard Check set when Bus Interconnection Check Stop not set	818	
0701		Hard Check not set when Bus Interconnection Check Stop set	825	

AC11 - MIOC CARD ERROR LATCH OUTPUT PATH TEST

This routine checks that each error latch in the CCU external to MIOC card sets the Hard Check bit in the CCU-to-MOSS Status A register.

1. Turn on, one-by-one, the series of latches.
2. Verify the CCU-to-MOSS Status A register.

ERC	Function	Error Description	RAC	Comments
0700	Read CCU-to-MOSS Status A register after setting an error latch in the CCU.	Only one latch failed to set Hard Check.	814 815 816 818 81A	
0701		All latches failed, give FRU in error.	815 81A 845 848 846 849 84A 84B 84C 84D 84E 84F 850 851	

AC12 - HARD CHECK 'SOFT' ERROR PATH TEST

This routine checks that soft error latches, gated by the 'Level 1 Entered' latch in the CCU-to-MOSS Status D register, set the Hard Check bit in the CCU-to-MOSS Status A register.

1. Set, via LSSD operations, the following latches: LVL5 I/O Error, Invalid Op, Address Exception on Instruction Fetch, Address Exception on Program Execution, Storage Protect on Instruction Fetch, and Storage Protect on Program Execution; set also the 'Level 1 Entered' latch.

2. Verify CCU-to-MOSS Status A register.

3. Turn off the 'Level 1 Entered' latch and turn on all the latches named above.

ERC	Function	Error Description	RAC	Comments
	Check CCU-to-MOSS Status A register, with 'level 1 entered' latch on			RCV DATA field gives the latches in error (see note)
0700		Hard Check not set in CCU-to-MOSS Status A register for all latches	811	
0701		Hard Check not set in CCU-to MOSS Status A register for some latches	81A 847	
702	Check CCU-to-MOSS Status A register, with 'level 1 entered' latch off	Hard Check set in CCU-to-MOSS Status A register	811	

Note: RCV DATA field byte description:

Bit	Description
0	(not used)
1	(not used)
2	LVL5 I/O error
3	Invalid op
4	Address exception on instruction fetch
5	Storage protect on instruction fetch
6	Address exception on program execution
7	Storage protect on program execution

AC13 - BYPASS CCU CHECK STOP LATCH TEST

Check that the 'bypass CCU check stop' bit set in the Diagnostic Mode Control register prevents the setting of the Hard Check condition. The same test is done for the 'Set/Reset Bypass Check Stop' latch on the BTAC card.

1. Set, via the LSSD, the MIOC Error latch and the Bypass CCU Check Stop latch.
2. Set, via the LSSD, Set/Reset Bypass CCU Check Stop latch on the BTAC card and the MIOC Error latch.
3. Verify twice that no Hard Check stop occurred.

ERC	Function	Error Description	RAC	Comments
0700	Set 'Bypass CCU Check Stop' latch on MIOC card.	CCU Hard Check occurred	80A	RCV DATA field gives contents of CCU-to-MOSS Status A register
0701	Set 'Bypass CCU Check Stop' latch on BTAC card.	CCU Hard Check occurred	812	

AC14 - PROGRAM STOP AND AIO STOP SETTING TEST

This routine checks that a CCU Hard Check causes the setting of the Program Stop and AIO Stop latches. The Bypass CCU Check Stop bit, if on, prevents the setting of these two latches.

1. In clock step mode, turn on the MIOC Error latch and turn on (or off) the Bypass CCU Check Stop latch.
2. Advance the clock one step.
3. Verify that the Program Stop and AIO Stop latches are on (or off) when Bypass CCU Check Stop is off (or on).

ERC	Function	Error Description	RAC	Comments
0700	Read Mode Control Register B with 'Bypass CCU Check Stop' latch on	Program Stop and AIO Stop latches on	811	
0701	Read Mode Control Register B with 'Bypass CCU Check Stop' latch off	Program Stop and AIO Stop latches off	811	

AC15 - CCU CHECK RESET FUNCTION TEST

This routine checks that the CCU Check Reset bit in the Diagnostic Mode Control register, when on, resets all the Hard Check error latches.

1. Turn on a series of error latches using LSSD operations and set the CCU Check Reset bit.

2. Verify the state of the latches.

ERC	Function	Error Description	RAC	Comments
0700	Read all error latches previously set.	All latches stay in the on state.	811	
0701		Only one error latch stays in the on state.	81A 814 815 816 818	
0702		Some error latches stay in the on state.	811 815 81A 845 846	
0703		Error latch set on MIOC card	811	

AC16 - MOSS INTERRUPT DISABLE FUNCTION TEST

This routine checks that the MOSS Interrupt Disable bit in the Diagnostic Mode Control register, when on, prevents a HLIR to the MOSS.

1. Via an LSSD operation set the MIOC Error latch and disable the HLIR latch.

2. Verify MCC Status Register 1 for a HLIR.

ERC	Function	Error Description	RAC	Comments
0700	Read MCC Status Register 1	HLIR not degated by the MOSS interrupt disable	811	

AD01 - ROSAR BYTE 0 PARITY CHECKER TEST

This routine checks that the parity checker on the ROSAR byte 0 detects parity errors, and propagates the error condition to the MIOC Error latch.

1. Execute a write using LSSD bit 0 and the parity bit of byte 0.
2. Write X'00' in the String Select register to enable the gating to the MIOC Error latch.
3. Read the MIOC LSSD string and check that the MIOC Error latch is on.

ERC	Function	Error Description	RAC	Comments
0700	Write value 0 in two bits of the ROSAR and check the MIOC Error latch.	MIOC Error latch not set	811	
0701	Write value 1 in two bits of the ROSAR and check the MIOC Error latch.	MIOC Error latch not set	811	

AD02 - ROSAR BYTE 1 PARITY CHECKER TEST

This routine checks that the parity checker on the ROSAR byte 1 detects parity errors, and propagates the error condition to the MIOC Error latch.

1. Write byte 1 of the ROSAR with 0.
2. Write X'00' in the String Select register to enable gating of the MIOC Error latch.
3. Read the MIOC LSSD string to verify the MIOC Error latch.
4. Repeat the test with several patterns in order to exercise all logic blocks in the parity checker.

ERC	Function	Error Description	RAC	Comments
0700	Write 0 to byte 1 of ROSAR and check MIOC Error latch	MIOC Error latch not set	811	
0701	Write different patterns to byte 1 of ROSAR and check MIOC Error latch	MIOC Error latch not set	811	

AD03 - LSAR PARITY CHECKER TEST

This routine checks that the parity checker on the LSAR detects parity errors, and propagates the error condition to the MIOC Error latch.

1. Write LSAR with 0.
2. Write 0 in the String Select register to enable gating of the MIOC Error latch.
3. Read the MIOC LSSD string to verify the MIOC Error latch.
4. Repeat the test with several patterns in order to exercise all logic blocks in the parity checker.

ERC	Function	Error Description	RAC	Comments
0700	Write 0 to LSAR and check MIOC Error latch.	MIOC Error latch not set	811	
0701	Write different patterns to LSAR with correct parity and check that the MIOC Error latch stays reset.	MIOC Error latch set	811	

AD04 - MODE CONTROL REGISTER A PARITY CHECKER TEST

This routine checks that the parity checker on Mode Control register A detects parity errors, and propagates the error condition to the MIOC Error latch.

1. Write the Mode Control Register A with 0.
2. Write 0 in the String Select register to enable gating of the MIOC error latch.
3. Read the MIOC LSSD string to verify the MIOC Error latch.
4. Repeat the test with several patterns in order to exercise all logic blocks in the parity checker.

ERC	Function	Error Description	RAC	Comments
0700	Write 0 to Mode Control Register A	MIOC Error latch not set	811	
0701	Write different patterns to Mode Control Register A with correct parity and check that the MIOC Error latch stays reset.	MIOC Error latch set	811	

AD05 - DIAGNOSTIC MODE CONTROL REGISTER PARITY CHECKER TEST

This routine checks that the parity checker on the Diagnostic Mode Control register detects parity errors, and propagates the error condition to the MIOC Error latch.

1. Write the Diagnostic Mode Control register with 0.
2. Write 0 in the String Select register to enable gating of the MIOC error latch.
3. Read the MIOC LSSD string to verify the MIOC Error latch.
4. Repeat the test with several patterns in order to exercise all logic blocks in the parity checker.

ERC	Function	Error Description	RAC	Comments
0700	Write 0 to Diagnostic Mode Control register and check the MIOC Error latch	MIOC Error latch not set	811	
0701	Write different patterns to the Diagnostic Mode Control register with correct parity and check that the MIOC error latch stays reset.	MIOC Error latch set	811	

AD06 - ADDRESS COMPARE CONTROL REGISTER PARITY CHECKER TEST

This routine checks that the parity checker on the Address Compare Control register detects parity errors, and propagates the error condition to the MIOC Error latch.

1. Write the Address Compare Control register with 0.
2. Write 0 in the String Select register to enable gating of the MIOC Error latch.
3. Read the MIOC LSSD string to verify the MIOC Error latch.
4. Repeat the test with several patterns in order to exercise all logic blocks.

ERC	Function	Error Description	RAC	Comments
0700	Write 0 to Address Compare Control register and check the MIOC Error latch.	MIOC Error latch not set	811	
0701	Write different patterns to the Address Compare Control register with correct parity and verify that the MIOC error latch stays reset.	MIOC Error latch set	811	

AE01 - MODE CONTROL REGISTER B TEST

This routine checks that the latches of the Mode Control Register B can be set and reset via MOSS direct read/write operations.

1. Read the register after CCU initialisation.
2. Turn off one bit at a time and verify.
3. Turn on one bit at a time and verify.
4. Reset each bit that was set and verify the reset.

ERC	Function	Error Description	RAC	Comments
0700	Test the register after CCU initialisation.	Invalid data in the Mode Control Register B	841	EXP DATA and RCV DATA fields contains
0701	Reset bits 0, 1, and 2 in Mode Control Register B.	Bits not reset	811	expected data and received data
0702	Set bits 0, 1, 2 in the Mode Control Register B	Bits not set	811	respectively

AE02 - DIAGNOSTIC MODE CONTROL REGISTER TEST

This routine checks that the latches of the Diagnostic Mode Control register can be set and reset via MOSS direct operations.

1. Write a pattern of floating ones and zeros.
2. Read the contents of the register using LSSD operations.
3. Compare the read/write data and check the MIOC Error latch.

ERC	Function	Error Description	RAC	Comments
	Write/read Diagnostic Mode Control register			EXP DATA and RCV DATA fields contain written
0701		Data Error only	811	data and read data
0702		Parity Latch Error		respectively
0703		Parity Error and Data Error		
0704		MIOC Error		
0705		MIOC Error and Data Error		
0706		MIOC Error and Parity Error		
0707		MIOC Error, Parity Error, and Data Error.		

AE03 - BRANCH TRACE LEVEL CONTROL REGISTER TEST

This routine checks that the latches of the Branch Trace Level Control register can be set and reset via MOSS direct operations.

1. Write a pattern of floating ones and zeros.
2. Read the contents of the register using LSSD operations.
3. Compare the read/write data and check the MIOC Error latch.

ERC	Function	Error Description	RAC	Comments
	Write/read the Branch Trace Level Control register.			EXP DATA and RCV DATA fields contains written data and read data respectively
0701		Data Error only	811	
0702		Parity Latch Error		
0703		Parity Error and Data Error		
0704		MIOC Error		
0705		MIOC Error and Data Error		
0706		MIOC Error and Parity Error		
0707		MIOC Error, Parity Error, and Data Error.		

AE04 - ADDRESS COMPARE CONTROL REGISTER TEST

This routine checks that the latches of the Address Compare Control register can be set and reset via MOSS direct operations.

1. Write a pattern of floating ones and zeros.
2. Read the contents of the register using LSSD operations.
3. Compare the read/write data and check the MIOC Error latch.

ERC	Function	Error Description	RAC	Comments
	Write/read Address Compare Control register			EXP DATA and RCV DATA fields contain written data and read data respectively
0701		Data Error only	811	
0702		Parity Latch Error		
0703		Parity Error and Data Error		
0704		MIOC Error		
0705		MIOC Error and Data Error		
0706		MIOC Error and Parity Error		
0707		MIOC Error, Parity Error, and Data Error.		

AE05 - MODE CONTROL REGISTER A TEST

This routine checks that the latches of the Mode Control Register A can be set and reset via MOSS direct operations.

1. Write a pattern of floating ones and zeros.
2. Read the contents of the register using LSSD operations.
3. Compare the read/write data and check the MIOC Error latch.

ERC	Function	Error Description	RAC	Comments
0701	Write/read Mode Control Register A	Data Error only	811	EXP DATA and RCV DATA fields contain written data and read data respectively
0702		Parity Latch error	811	
0703		Parity error and Data Error		
0704		MIOC Error		
0705		MIOC Error and Data Error		
0706		MIOC Error and Parity Error		
0707		MIOC Error, Parity Error and Data Error		

AE06 - LSAR TEST

This routine checks that the latches of the LSAR can be set and reset via MOSS direct operations.

1. Write a pattern of floating ones and zeros.
2. Read the contents of the register using LSSD operations.
3. Compare the read/write data and check the MIOC Error latch.

ERC	Function	Error Description	RAC	Comments
0701	Write/read Mode Control Register A	Data Error only	811	EXP DATA and RCV DATA fields contain written data and read data respectively
0702		Parity Latch Error	811	
0703		Parity Error and Data Error		
0704		MIOC Error		
0705		MIOC Error and Data Error		
0706		MIOC Error and Parity Error		
0707		MIOC Error, Parity Error and Data Error		

AE07 - ROSAR BYTE 0 TEST

This routine checks that the latches of the ROSAR (byte 0) can be set and reset via MOSS direct operations.

1. Write a pattern of floating ones and zeros.
2. Read the contents of the register using LSSD operations.
3. Compare the read/write data and check the MIOC Error latch.

ERC	Function	Error Description	RAC	Comments
	Write/read the ROSAR byte 0			EXP DATA and RCV DATA fields
0701		Data Error only	811	contains written data and read data respectively
0702		Parity Latch Error		
0703		Parity Error and Data Error		
0704		MIOC Error		
0705		MIOC Error and Data Error		
0706		MIOC Error and Parity Error		
0707		MIOC Error, Parity Error and Data Error		

AE08 - ROSAR BYTE 1 TEST

This routine checks that the latches of the ROSAR (byte 1) can be set and reset via MOSS direct operations.

1. Write a pattern of floating ones and zeros.
2. Read the contents of the register using LSSD operations.
3. Compare the read/write data and check the MIOC Error latch.

ERC	Function	Error Description	RAC	Comments
	Write/read the ROSAR byte 1			EXP DATA and RCV DATA fields
0701		Data Error only	811	contains written data and read data respectively
0702		Parity Latch Error		
0703		Parity Error and Data Error		
0704		MIOC Error		
0705		MIOC Error and Data Error		
0706		MIOC Error and Parity Error		
0707		MIOC Error, Parity Error and Data Error		

AE09 - MOSS-TO-CCU STATUS REGISTER TEST

This routine checks that the latches of the MOSS-to-CCU Status register can be set and reset via MOSS direct operations.

1. Write a pattern of floating ones and zeros.
2. Read the contents of the register using LSSD operations.
3. Compare the read/write data and check the MIOC Error latch.

ERC	Function	Error Description	RAC	Comments
	Write/read the MOSS-to-CCU Status register			EXP DATA and RCV DATA field contain written and read data respectively
0701		Data Error only	811	
0704		MIOC Error only		
0705		Data and MIOC Error		

AE10 - CCU-TO-MOSS STATUS E REGISTER TEST

This routine checks that the latches of the MOSS-to-CCU Status E register can be set and reset.

1. After CCU initialization, read back the CCU-to-Moss Status E register via an indirect operation.
2. Write a pattern of floating 1s and 0s and read it back.
3. Compare the read/write data and check the MIOC Error latch.

ERC	Function	Error Description	RAC	Comments
0700	Read the register after initialization	One latch set after initialization	80C	
0701	Write/read the register with different patterns	One latch not set or reset	811	

AE12 - MDOR CONTROL LINE TEST

This routine checks that the MDOR control lines (A, B, C, and D) when set to X'xxx0', are decoded to enable the correct latch output to be gated to the MOSS bus.

1. Set MDOR byte X to X'FF' and set the ROS MDOR control latches (RMDO-14, RMDO-15, RMDO-16, and RMDO-17) to 100, 010, 001, and 000 respectively.
2. Read direct address X'00'.
3. Test the read data.
4. Repeat steps 1 through 3 for bytes 0 and 1 of the MDOR.

ERC	Function	Error Description	RAC	Comments
0700	Test data for X'FF'	Data always X'00'	819	Error displays: ADDIT INFO field contains 'MDOR READ ERROR' followed by the 3-byte value of MDOR
0701	Test data for X'FF'	Some bits in error	861 862 863 864 865 866 867	

AE13 - MDOR CONTROL LINE TEST FROM MIOC

This routine checks that the MDOR control lines (A, B, C, and D) are set by the MIOC address decoder when MDOR bytes X, 0, and 1 are read by a direct read operation, and that the MDOR latches are gated to the MOSS bus.

1. Set MDOR byte X to X'FF' and bytes 0 and 1 to X'00' via LSSD operations.
2. Read byte X.
3. Verify that byte X contains X'FF' and check the MIOC Error latch.
4. Repeat steps 1 through 3 for bytes 0 and 1 of the MDOR (contents should be X'00').

ERC	Function	Error Description	RAC	Comments
0700	Verify data from MDOR and MIOC Error latch	Data is wrong but MIOC Error latch is not set	811	Error displays: ADDIT INFO field contains 'MDOR READ ERROR' followed by the 3-bytes value of MDOR
0701	Verify data from MDOR and MIOC Error latch	Some bits in error, and MIOC Error latch is set	861 862 863 864 865 866 867	

AE14 - MDOR DIRECT WRITE TEST

This routine checks that the MDOR Direct Write operation works correctly.

1. Initialize MDOR bytes X, 0, and 1 with X'00' via LSSD operations.
2. Write one byte with X'FF' and read the MDOR via LSSD operations.
3. Verify the data and parity bits.
4. Repeat for the other two bytes of the MDOR.
5. Restart the initialization of MDOR bytes X, 0, and 1 using X'FF' and test for X'00' patterns.

ERC	Function	Error Description	RAC	Comments
	Verify data bits and parity bits of MDOR			Error displays: ADDIT INFO field contains 'MDOR READ ERROR' followed by the 3-byte value of MDOR
0700		Errors in data bits and parity bits in all tests	811	
0701		Data bits are correct but all parity bits are wrong	861 862 863 864 865 866 867	
0702		Some data bits and/or parity bits are in error	861 862 863 864 865 866 867	

AE16 - MDOR PARITY BIT DRIVERS - PARITY CHECKER TEST

This routine checks that the parity checker for the MDOR bytes detects parity errors on the MDOR to MOSS bus from the CCU DFL1 cards. Test that the MDOR Parity Error latch is set by a parity error.

1. Set the parity bit of a byte in the MDOR to '0' and read the MDOR via a direct operation.
2. Verify that the MDOR Parity Error latch and the MIOC Error latch are set.
3. Repeat for the other two MDOR bytes.
4. Write different patterns to byte X of the MDOR via direct operations.
5. Read MDOR byte X via a direct operation and test the MDOR Parity Error latch.

ERC	Function	Error Description	RAC	Comments
0700	Verify MDOR Parity Error latch	MDOR Parity Error latch is wrong for all 3 bytes	811	Error display: ADDIT INFO field contains 'MDOR READ ERROR' followed by the 3-byte value of MDOR
0701	Verify MDOR Parity Error latch	MDOR Parity Error latch is wrong for 1 or 2 bytes	811	
0703	Verify MDOR Parity Error latch setting as a function of written data pattern	MDOR Parity Error latch not correctly set	811	

AE17 - CCU-TO-MOSS STATUS D REGISTER TEST

This routine checks that the latches of the MOSS-to-CCU Status D register can be set and reset.

1. Initialize the register via an LSSD operation, and read it back via an indirect operation.
2. Write a pattern of floating ones and zeros.
3. Compare the read/write data.
4. Using the gates ENTE-14, 15, and 17, and MASK-17, set the Program Wait State bit (bit 1).

ERC	Function	Error Description	RAC	Comments
0700	Read the register after initialisation	One latch set after initialisation	847	EXP DATA and RCV DATA fields contain written data and read data respectively
0701	Write/read register with different patterns	Latch not set or reset	841	
0702	Test Program Wait state latch	Latch not set	811	

AE18 - CCU-TO-MOSS STATUS F REGISTER TEST

This routine checks that the latches of the MOSS-to-CCU Status F register can be set and reset.

1. Initialize the register via an LSSD operation and read it back using a direct operation.
2. Write a pattern of floating ones and zeros.
3. Compare the read/write data.

ERC	Function	Error Description	RAC	Comments
0700	Read the register after initialization	A latch was on after initialization	80C	EXP DATA and RCV DATA fields contain written data and read data respectively
0701	Write/read register	Latch not set or reset	811	

AF01 - ELEMENTARY TRANSFER PATH TEST

This routine checks the data transfer:

- From the IAR to the LAR.
- From the IAR to the IAR via the IAR incrementer.

ERC	Function	Error Description	RAC	Comments
	IAR and corresponding gates (CT21-CT29, IPFC, BREG, RIAR, POPA) are loaded via LSSD operations. Do one clock step. 3 different value are used as IAR data: X'073AAAAA' X'07055555' X'00313131'			EXP DATA and RCV DATA fields contain correct and bad values respectively
0700	Verify contents of IAR	IAR does not contain value of the IAR+2.	8A3 8A4 8A5 8A6 8A7 8A8 8A9	
0701	Verify contents of LAR	LAR does not contain value of the IAR	829 82A 82B 82C 82D 82E 82F	

AF02 - ELEMENTARY TRANSFER PATH TEST

This routine checks the data transfer from the MDOR to the WSDR via the N-Bus Out.

ERC	Function	Error Description	RAC	Comments
	MDOR and corresponding gates (RMDO, RWSD, and MRST) are loaded via LSSD operations Do one clock step 3 different values are used as MDOR data: X'073AAAAA' X'07055555' X'00313131'			EXP DATA and RCV DATA fields contain correct and bad values respectively
0700	Verify that WSDR contains data from MDOR	WSDR does not contain the same data as MDOR	8A3 8A4 8A5 8A6 8A7 8A8 8A9	

AF03 - ELEMENTARY TRANSFER PATH TEST

This routine checks the gating of each POP to the OPDB and also checks for correct data transmission.

ERC	Function	Error Description	RAC	Comments
	Routine loads POPA, POPB, POPC, POPD with the data. Set IPFC and CT21-CT29 to control POP selection via LSSD operations. Do one clock step. Contents of OPDB is verified			EXP DATA and RCV DATA fields contain correct and bad values respectively
0700		OPDB does not contain data from POP	83A 83B 83C 83D 83E 83F 852	

AF04 - ELEMENTARY TRANSFER PATH TEST

This routine checks the data transfer:

- From the PFAR or the SAR.
- From the SAR to the SAR via the SAR incrementer.

ERC	Function	Error Description	RAC	Comments
	Routine loads IPFC and PFAR. Do one clock step			EXP DATA and RCV DATA fields contain correct
0700	Contents of SAR is verified.	SAR does not contain data from PFAR.	8AA 8AB 8AC 8AD 8AE 8AF 874	and bad values respectively
	SAR is loaded with data. SAR incrementer is gated via LSSD operations (RWSA, RSAS, and RSIN). Contents of SAR is incremented for 0, 1, 2, 4 bytes. PFAR and SAR are loaded with 3 different values: X'073AAAAA' X'07055555' X'00313131'		8B0 8B1 8B2 8B3 8B3 8B4 8B5	
0701	Contents of SAR is verified	SAR is not incremented	8B6	

AG01 - ELEMENTARY TRANSFER PATH TEST

This routine checks that the Z register is not erroneously modified when no Z bus source is gated to the Z bus.

1. Reset all the gates to the Z bus (RZBS, RSTZ, RSOE, STUI, RALX, RAL0, RAL1 and MIS3) and set all the bits of the Z register to 1.
2. Do one clock step.
3. Verify that the Z register stays at its initial value.
4. Repeat the test with the Z register initialized to 0.
5. If the previous test failed, the most probable source of the error is the LAR. Initialize the LAR to 0 and repeat the test for a 0 value in the Z register.

ERC	Function	Error Description	RAC	Comments
0700	Verify the contents of the Z register.	Z register does not contain all 1s.	846 8B7 8B8 8B9 8BA 8BB 8B0	For ERC 0700 correct and bad values are displayed on the screen in EXP DATA and RCV data fields respectively
0701	Verify the contents of the Z register for error coming from the LAR after the test for the 0 value failed.	Z register is correct Z register is still wrong	816 874	

AG02 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the IAR and Work registers to the Z register.

ERC	Function	Error Description	RAC	Comments
	- Load IAR and WRK1-WRK7 one by one with 3 different patterns: X'073AAAAA' X'07055555' X'00313131' - Load gates RZBS, and RSTZ via LSSD operations. - Initiate LSAR for each register - Do one clock step - Verify contents of Z register.			Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0700	IAR --> Z register	All bits of the Z register are wrong	874	
0701	WRKI --> "			
0702	WRK2 --> "			
0703	WRK3 --> "	Some bits of the Z register are wrong	846	
0704	WRK4 --> "		8B7	
0705	WRK5 --> "		8B8	
0706	WRK6 --> "		8B9	
0707	WRK7 --> "		8BA 8BB 8BC	

AG03 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the LAR to the Z register.

ERC	Function	Error Description	RAC	Comments
	Load LAR with 3 different patterns: X'07BAAAAA' X'07055555' X'00313131' Initialize LAR to Z Bus gate (RZBS) via LSSD operation Do one clock step			Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0700	Verify contents of Z register	Contents of Z register is not equal to LAR	87F	

AG04 - ELEMENTARY TRANSFER PATH TEST

This routine checks the following transfers:

- From the OPDB (bits 1.2 through 1.7) to the Z register (bits X.2 through X.7).
- From the POPA (bytes 0 and 1) to the Z register (bytes 0 and 1).

ERC	Function	Error Description	RAC	Comments
	Load gates via LSSD operations (CT21-29, RSTZ, RZBS) Load OPDB with 3 different values: X'02FFFA' X'02FFC5' X'02FFF1' Load POPA with 3 different values: X'03AAAA' X'035555' X'003131' Do one clock step			Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0700	Verify contents of the Z register which should contain: X'033AAAAA' X'03055555' X'00313131'	All Z register bits are wrong Some Z register bits are wrong	874 84C 8B7 8B8 8B9 8BA 8BB 8BC	

AG05 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the SAR to the Z register.

ERC	Function	Error Description	RAC	Comments
	Load SAR with 3 different patterns: X'073AAAAA' X'07055555' X'00313131' Initialize RZBS and RSTZ gates via LSSD operations. Do one clock step			Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0700	Verify contents of the Z register.	All Z register bits are wrong	874	
		Some Z register bits are wrong	84C 8B7 8B8 8B9 8BA 8BB 8BC	

AH01 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the Z register to the IAR and to the Work registers.

1. Write the Z register via an LSSD operation.
2. Set the gates for a transfer to the IAR (RIAR, R1WW) or to the Work registers (RWKS, RPTY, RIMM, RZBS) via LSSD operations.
3. Do one clock step.
4. Verify the contents of the IAR and WRK1-7.

Note: Each path is tested with 3 values: X'07BAAAAA', X'07055555', and X'00313131'.

ERC	Function	Error Description	RAC	Comments
0700	Test path Z register to IAR	All bits in error	874	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	" WRK1			
0702	" WRK2			
0703	" WRK3	Some bits in error	813	
0704	" WRK4		814	
0705	" WRK5		815	
0706	" WRK6		8C6	
0707	" WRK7		8C7	
			8C8	
			875	

AH02 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the Z register to the MDOR.

1. Write the Z register via an LSSD operation.
2. Set the gates for a transfer to the MDOR (RMDO, RZBS), via LSSD operations.
3. Do one clock step.
4. Verify the MDOR contents.

Note: The path is tested with 3 values: X'073AAAAA', X'07055555', and X'00313131'.

ERC	Function	Error Description	RAC	Comments
0700	Test path Z register to MDOR	All bits in error	874	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
		Some bits in error	813	
			814	
			815	
			8C6	
			8C7	
			8C8	
			875	

AH03 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the Z register to the SAR.

1. Write the Z register via an LSSD operation.
2. Set the gates for a transfer to the SAR (RSAS, RZBS, RPTY, and RWSA) via LSSD operations.
3. Do one clock step.
4. Verify the SAR contents.

Note: The path is tested with 3 values: X'073AAAAA', X'07055555' and X'00313131'.

ERC	Function	Error Description	RAC	Comments
0700	Test path Z register to SAR	All bits in error	874	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
		Some bits in error	813 814 815 8C6 8C7 8C8 875	

AH04 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the Z register to the WSDR.

1. Write the Z register via an LSSD operation.
2. Set the gates for a transfer to the WSDR (RWSD, RZBS) via LSSD operations.
3. Do one clock step.
4. Verify the contents of the WSDR.

Note: The path is tested with 3 values: X'073AAAAA', X'07055555' and X'00313131'.

ERC	Function	Error Description	RAC	Comments
0700	Test path Z register to WSDR	All bits in error	874	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
		Some bits in error	813 814 815 8C6 8C7 8C8 875	

AH05 - ELEMENTARY TRANSFER PATH TEST

This routine checks the following transfers:

- From the WSDR to the Z bus.
- From the WSDR to the storage bus.
- From the WSDR to N-Bus In.

1. Write the WSDR via an LSSD operation.
2. Set the gates for a transfer to the Z register (STUI, RSOE, RSTZ, RPTY, MIS3, and SAR) via LSSD operations.
3. Do one clock step.
4. Verify the Z register contents.
5. Write the WSDR via an LSSD operation.
6. Set the gates for a transfer to the POPX (IPFC, RPTY, STUI, RSOE, MIS3) via LSSD operation.
7. Do one clock step.
8. Verify the POPA, POPB, POPC and POPD contents.
9. Write the WSDR via an LSSD operation.
10. Set the gates for a transfer to N-Bus In (RMDO, STUI, RSOE, and RPTY) via LSSD operations.
11. Do one clock step.
12. Verify the MDOR contents.

Note: Each path is tested with 3 values: X'073AAAAA', X'07055555', and X'00313131'.

ERC	Function	Error Description	RAC	Comments
0700	Test WSDR to Z bus transfer	All bits in error for both transfer paths. Some bits in error in WSDR to Z bus XFER.	8CA 80D 80E 877 8ED	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701 0702 0703 0704	Test WSDR to POPA)storage Test WSDR to POPB) bus Test WSDR to POPC)transfer Test WSDR to POPD)	Some bits in error in WSDR to storage bus transfer	8F0	
0705	Test WSDR to N-Bus In	Bits in error in MDOR	813 814 815 8C6 8C7 8C8 875	

AI01 - ELEMENTARY TRANSFER PATH TEST

This routine checks the following transfers:

- From the IAR to the Z register via the A-Bus and the ALU.
- From the WRK1-WRK7 to the Z register via the A-Bus and the ALU.
- From the WSDR to the Z register via the A-Bus and the ALU.

1. Initiate the IAR, WRK1-WRK7 and WSDR successively with 3 different patterns, via LSSD operations.
2. Set the gates for a transfer to the A-Bus (CT11-CT19, CT21-CT29, RALX, RAL0, and RAL1) via LSSD operations.
3. Do one clock step.
4. Verify the contents of the Z register. Set a flag for each incorrect verification.

Note: The patterns used are X'073AAAAA', X'07055555' and X'00313131'.

ERC	Function	Error Description	RAC	Comments
	Verify contents of the Z register after the transfer from:	All transferred bits in error	8BD to 8C4	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0700	IAR			
0701	WRK1			
0702	WRK2			
0703	WRK3			
0704	WRK4	Some transferred bits in error	875	
0705	WRK5		887	
0706	WRK6		to	
0707	WRK7		88C	
0708	WSDR			

AI02 - ELEMENTARY TRANSFER PATH TEST

This routine checks the following transfers:

- From the IAR to the Z register via the B-Bus and the ALU.
- From the WRK1-WRK7 to the Z register via the B-Bus and the ALU.
- From the WSDR to the Z register via the B-Bus and the ALU.

1. Initiate the IAR, WRK1-WRK7 and OPDB successively with 3 different patterns, via LSSD operations.
2. Set the gates for a transfer to the B-Bus (CT11-CT19, RALX, RAL0, and RAL1) via LSSD operations.
3. Do one clock step.
4. Verify the contents of the Z register. Set a flag for each incorrect verification.

Note: The patterns used are X'073AAAAA', X'07055555' and X'00313131'.

ERC	Function	Error Description	RAC	Comments
0700 0701 0702 0703 0704 0705 0706 0707 0708	Verify contents of the Z register after the transfer from IAR WRK1 WRK2 WRK3 WRK4 WRK5 WRK6 WRK7 OPDB	All transferred bits in error Some transferred bits in error	8BD to 8C4 875 887 to 88C	Correct and bad values are displayed in the screen in EXP DATA and RCV DATA fields respectively

AI03 - ELEMENTARY TRANSFER PATH TEST

This routine checks the following transfers:

- From the IAR to the Z register via the B-Bus and the ALU.
- From THE WRK1-WRK7 to the Z register via the B-Bus and the ALU.
- From the WSDR to the Z register via the B-Bus and the ALU.

1. Initiate the IAR, WRK1-WRK7 and OPDB successively with 3 different patterns, via LSSD operations
2. Set the gates for a transfer to the B BUS CROSS (CT11-CT19, RALX, RAL0, and RAL1) via LSSD operations.
3. Do one clock step.
4. Verify the contents of the Z register. Set a flag for each incorrect verification.

Note: The patterns used are X'073AAAAA', X'07055555' and X'00313131'.

ERC	Function	Error Description	RAC	Comments
	Verify contents of the Z register after the transfer from:	All transferred bits in error	8BD to 8C4	Correct and bad values are displayed in the screen in EXP DATA and RCV DATA fields respectively
0700	IAR			
0701	WRK1			
0702	WRK2			
0703	WRK3			
0704	WRK4	Some transferred bits in error	875	
0705	WRK5		887	
0706	WRK6		to	
0707	WRK7		88C	
0708	OPDB			

AI04 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the Z1-Bus to the Z-Bus.

1. Initiate the latches of the In X'76'.
2. Write the Op register to simulate the In X'76'.
3. Set the gates (RSTR, MIS1, OPCC).
4. Do one clock step.
5. Verify the Z reg.
6. Repeat the same with the latches set to zero, so that the Z register contains X'00' after the In X'76'.
7. Do the same for the In X'7D'.

ERC	Function	Error Description	RAC	Comments
0700	Test Z register for In X'76' or In X'7D'.	Reg does not contain In '76' or In X'7D'.	80F	

AJ01 - CCU SRL TEST WITH INITIAL VALUE (MANUALLY INVOKED ROUTINE)

This routine checks that all the LSSD latches are set to their initial value after Scan-In and C-Clock have been started.

1. Scan-In the initial value of all LSSD strings.
2. Start the C-Clock.
3. Scan-Out all the LSSD strings
4. Compare the written and read data.

ERC	Function	Error Description	RAC	Comments
0700	Read String Select register to check C-Clock step	String Select register not equal to 0	811	To interpret ERR BIT field and ADDIT INFO field see MIM2 "CCU Extended Replacement Procedure"
0701	Display FRUs with number of errors by FRU		8FD	
0702	Display latches in error for the FRU with highest number of errors.		8FD	

AK01 - ROS CONTENTS TEST

This routine checks all the ROS words for correct contents.

1. Select all 512 words in turn from the ROSAR.
2. Do one clock step.
3. Verify the contents of the latches.

ERC	Function	Error Description	RAC	Comments
0701	Read, using LSSD operations, the latches set by every ROS word and compare with the expected values provided by a table.	One or more latches incorrectly set.	85D 852 82F 819 87F 877 82F 85B 85C 873 871 826	

AK02 - ROS ADDRESSING CONTROL TEST

Verify that a correct ROS word is selected from the POP decode.

1. Write an instruction in the POP register
2. Prepare the IPF Control latches and reset the Program Stop latch.
3. Verify that a correct ROS word was selected.

ERC	Function	Error Description	RAC	Comments
0701	Read, using LSSD operations, the latches set by a particular ROS word	Wrong ROS word was selected	819	

AK03 - ROS WORD CHAINING TEST

Verify that ROS words are correctly chained.

1. Select ROS word X'1B'.
2. In the step mode, verify that all 8 chained words are correctly selected.

ERC	Function	Error Description	RAC	Comments
0701	Read, using LSSD operations, the latches set by all 8 chained ROS words	An error occurred in the selection of the next ROS word.	819	

BA01 - ADDRESS COMPARE ADDRESS 1 AND ADDRESS 2 REGISTER TEST

ERC	Function	Error Description	RAC	Comments
0700	Address Compare Address 1 register is written via an indirect operation with floating 0 and floating 1 patterns. Each pattern is read back via LSSD ops. Written and read values are compared.	Written and read values are different	853	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Repeat using the Address Compare Address 2 register	Written and read values are different.	871	

BA02 - BRANCH TRACE LOWER AND UPPER LIMIT REGISTERS TEST

ERC	Function	Error Description	RAC	Comments
0700	Branch Trace Lower Limit register is written via an indirect operation with floating 0 and floating 1 patterns. Each pattern is read back via LSSD ops. Written and read values are compared.	Written and read values are different	853	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Repeat for the Branch Trace Upper Limit register.	Written and read values are different.	871	

BA03 - LOCAL STORE ADDRESSING TEST

ERC	Function	Error Description	RAC	Comments
0700 to 077F	Write local store from address X'00' to X'7F' with data patterns equal to the Local Store address. Read Local Store from address X'00' to X'7F' via indirect operations. Check contents of read LS.	Written and read values are different.	828	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BA04 - LOCAL STORE DATA SENSITIVITY

ERC	Function	Error Description	RAC	Comments
0700 to 077F	Local Store address X'00' to X'7F' is written with floating zeros and floating 1 patterns. Each pattern is read back. Written and read patterns are compared.	Written and read values are different.	873	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BA05 - LOCAL STORE INITIALISATION

ERC	Function	Error Description	RAC	Comments
0740 to 0746	Write Local Store X'40' to X'46' with initial values: LS40: X'0010' LS41: X'0080' LS42: X'0100' LS43: X'0180' LS44: X'0680' LS45: X'0700' LS46: X'0780' Read back LS X'40' to X'46' and compare result.	Written and read data values are different. The 2 last digits of ERC give the Local Store address in error.	873	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BB01 - ELEMENTARY TRANSFER PATH TEST

This routine checks the following transfers:

- From Local Store to the MDOR.
- From Local Store to the WSDR.
- From Local Store to the IAR.
- From Local Store to the SAR.

ERC	Function	Error Description	RAC	Comments
0700	Using ROS word X'9C' data is transferred from LS X'74' to the MDOR in clock step mode. 3 data patterns are used: X'073AAAAA' X'07055555' X'00313131' After advancing the C-Clock, the CCU User Indicator and CCU-to-MOSS Status A reg is verified.	Data in MDOR is different to LS X'74' or CCUI is not 02 (MOSS) or Hard Check occurred.	88F 890 891 892 893 894 8A1 8FF	Correct and bad values are displayed on the screen in EXP DATA and RCD DATA fields respectively
0701	Using ROS word X'17' data is transferred in the same way to the WSDR.	Data in WSDR is different to LS X'74' or CCUI problem or Hard Check occurred.		
0702	Using ROS word X'19' data is transferred in the same way to the IAR.	Data in IAR is different to LS X'74' or CCUI problem or Hard Check occurred.		
0703	Using ROS word X'65' data is transferred the same way to the SAR.	Data in SAR is different to LS X'74' or CCUI problem or Hard Check occurred.		

BB02 - ELEMENTARY TRANSFER PATH TEST

This routine checks the following transfers:

- From Local Store to the IAR
- From Local Store to WRK1-WRK7

ERC	Function	Error Description	RAC	Comments
0700 to 0707	Using ROS word X'1B' data is transferred from LS X'74' to IAR and WRK1-WRK7. 3 data patterns are used: X'073AAAAA' X'07055555' X'00313131'	Data in one of tested registers is different to LS X'74'. ERC indicates the failing register. 700: IAR 701: WRK1 702: WRK2 703: WRK3 704: WRK4 705: WRK5 706: WRK6 707: WRK7	88F 890 891 892 893 894 8A1	Correct and bad values are displayed on the screen in EXP DATA and RCD DATA fields respectively
	After advancing C-Clock, the CCU User Indicator and CCU to MOSS Status A register are verified.	User is not MOSS, or Hard Check bit is on	8FF	

BB03 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the Local Store to the IAR via the IAR incrementer.

ERC	Function	Error Description	RAC	Comments
0700	Using ROS words X'63' and X'64' data is tranferred from LS X'74' to IAR via IAR incrementer Four data patterns are used: X'073AAAAA' X'07055555' X'01313131' X'031FFFFF'	Data in IAR is not the data in LS X'74' incremented by 2	896 897 898 899 89A 89B 89C	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
	After advancing C-Clock, the CCU user indicator and CCU-to-MOSS Status A register are verified	User is not MOSS, or Hard Check bit is on	8FF	

BB04 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer from the Local Store to the SAR via the SAR incrementer (+0, +1, +2, and +4).

ERC	Function	Error Description	RAC	Comments
0700	Using ROS word X'67' data is transfered from LS X'74' to SAR via SAR incrementer 0. 4 data patterns are used: X'073AAAAA' X'07055555' X'00313131' X'031FFFFF' After advancing C-Clock, CCUI and CCU-to-MOSS Status A Registers are verified.	SAR is incorrectly incremented.	896 897 898 899 89A 89B 89C	Correct and bad values are displayed via EXP DATA and RCV DATA fields respectively
0701	Using ROS word X'69' data is transferred from LS X'74' to SAR via SAR incrementer 1.			
0702	Using ROS word X'6B' data is transferred from LS X'74' to SAR via SAR incrementer 2.	User is not MOSS or Hard Check bit is on	8FF	
0703	Using ROS word X'6D' data is transferred from LS X'74' to SAR via SAR incrementer 4.			

BB05 - ELEMENTARY TRANSFER PATH TEST

This routine checks the transfer path for all ALU operations.

ERC	Function	Error Description	RAC	Comments
0700 to 070D	Using ROS words, data is moved from LS X'74'-X'75' to IAR and WSDR (A and B bus). Next C-Clock step initiates one of 14 ALU operations. All ALU operations are tested. Result is read in LS X'76'	Incorrect value in LS X'76' after ALU operation.	887 888 889 88A 88B 88C 875	Correct and bad values are displayed via EXP DATA and RCV DATA fields respectively
	After advancing C-Clock CCUI and CCU-to-MOSS Status A register is verified.	700 Degate (all 1 with good parity) 701 A+C 702 Pass A and force carry out 703 A+not B+C (carry in = 1) 704 A + all 1 + C 705 Pass B 706 0 707 Shift B right enter shift in bit 708 Shift B right enter 0 709 A XOR B 70A A OR B 70B A AND B 70C Pass A 70D A+B+C	8FF	
		User is not MOSS or Hard Check bit is on.	8FF	

BC01 - LAR, IOC1-A, IOC1-D, IOC2-A, IOC2-D, AND OPDB REGISTERS READ TEST

ERC	Function	Error Description	RAC	Comments
0700 to 0705	LAR, IOC1-A, IOC2-A, IOC1-D, IOC2-D, and OPDB are written via LSSD operations with 3 different patterns: X'73555555' X'070AAAAA' X'001F0707' They are read back via indirect operations and contents are compared.	Invalid contents of one register: 700 LAR operation on data 701 IOC1-A operation on data 702 IOC1-D operation on data 703 IOC2-A operation on data 704 IOC2-D operation on data 705 OPDB	87F 887 888 889 88A 88B 88C 875 875	Correct and bad values are displayed on the screen in EXP DATA and RCV data fields respectively

BC02 - TEST OF IPF CONTROL

ERC	Function	Error Description	RAC	Comments
0700 to 0708	POPA, POPB, POPC, and POPD registers are loaded with 3 different patterns X'035555' X'03AAAA' X'000707' IPFC is initialized to all possible values one-by-one (04, 05, 06, 07, 08, 09, 0A, 0B, 00). Contents of OPDB is verified	Incorrect value in OPDB. ERC gives the IPFC value which causes the error: 700 04 701 05 702 06 703 07 704 08 705 09 706 0A 707 0B 708 00	887 888 889 88A 88B 88C 875	Correct and bad values are displayed on the screen and EXP DATA and RCV DATA fields respectively

BC03 - WRK1-WRK7, IAR AND SAR REGISTERS WRITE TEST

ERC	Function	Error Description	RAC	Comments
0700 to 0708	WRK1-WRK7, IAR, and SAR regs are written via indirect operations and read via LSSD operations. Written and read data are compared. 3 different patterns are used for each register: X'355555' X'0AAAAA' X'1F0707'	Written and read data are different. ERC gives failing register: 700 WRK1 701 WRK2 702 WRK3 703 WRK4 704 WRK5 705 WRK6 706 WRK7 707 IAR 708 SAR	887 888 889 88A 88B 88C 875	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BD01 - WORK1-WORK7, IAR AND SAR REGISTERS READ TEST

ERC	Function	Error Description	RAC	Comments
0700 to 0708	WORK1-WORK7, IAR, and SAR registers are written via indirect operations. They are read via indirect operations. Written and read values are compared 3 patterns are used for each register: X'355555' X'0AAAAA' X'1F7F7F'	Written and read values are different. ERC gives failing register: 700 WRK1 701 WRK2 702 WRK3 703 WRK4 704 WRK5 705 WRK6 706 WRK7 707 IAR 708 SAR	887 888 889 88A 88B 88C 875	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields respectively

BD02 - WSDR REGISTER TEST

ERC	Function	Error Description	RAC	Comments
0700	Write the WSDR via LSSD operations with C-Clock stopped. Do one clock step. Read WSDR via LSSD ops. and compare results. 3 patterns are used: X'07355555' X'070AAAAA' X'001F7F7F'	Written and read values are different.	887 888 889 88A 88B 88C 875	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields respectively

BD03 - WRAP BRANCH TRACE MECHANISM TEST

Initialize a branch trace in the wrap mode and test for branch trace count = 0 (end of branch trace buffer); the next branch trace buffer address = initial value, and branch trace buffer count = initial count.

ERC	Function	Error Description	RAC	Comments
	Initiate local storage: LS X'7B' with X'005555' LS X'7C' with X'00AAA2' LS X'7D' with X'001000' Force Branch Trace Counter LSDA to 0 via LSSD ops. Activate wrap branch trace mode.		871	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields respectively
0700	Read back LS X'7B' (current address) and verify that it is equal to X'7D' (buffer first address).	LS X'7B' does not contain X'1000'.		
0701	Read back LSDA counter	LSDA does not contain X'AAA0'.		

BD04 - IN X'70' TEST

Check the In X'70' instruction by comparing the execution result with the storage size value located in the CDF, and the hardware level indication set in MOSS storage at IML time.

ERC	Function	Error Description	RAC	Comments
0700	Issue indirect operation In X'70' and compare the value with contents of CDF.	Values do not match.	876	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields
0701	Issue an In X'70' and compare the hardware level indicator with the one saved in MOSS storage at IML time.	Values do not match.	876	Read In X'70' value displayed in RCV DATA field. EXP DATA field contents is meaningless

BD05 - IN X'73' TEST (INITIAL VALUE)

This routine checks the initial value of the keys (should be 0) for all possible key types in the first 4K of storage.

ERC	Function	Error Description	RAC	Comments
	Issue In X'73' with following modifiers for all possible keys.			Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0700	Modifier 000 = user key			
0701	Modifier 100 = user key + 2K block			
0702	Modifier 001 = storage prot. key	One of the keys is not equal to 0.	876	
0703	Modifier 101 = storage prot. key + 2K block			
0704	Modifier 011 = read only key			
0705	Modifier 111 = read only key + 2K block			
0706	Modifier 010 = addr. except			
0707	Modifier 110 = addr. except + 2K block			

BD06 - IN X'73' TEST (USER KEY)

This routine tests the user key data register.

ERC	Function	Error Description	RAC	Comments
0700	The UKDR register is written via LSSD operations and read via an In X'73' instruction. Written and read values are compared. All 8 possible values are used (0-7).	Written and read values do not match.	81A	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields respectively

BD07 - IN X'73' TEST (READ ONLY KEY)

This routine tests the read only key in the Storage Key Data register test.

ERC	Function	Error Description	RAC	Comments
0700	Read only key is written via LSSD for address 0 and 2K. Read only key is read via In X'73'. Written and read values are compared. Both 0 and 1 values are used.	Mismatch during compare	81A	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BD08 - IN X'73' TEST (STORAGE PROTECT KEY)

This routine tests the storage protect key in the Storage Key Data register test.

ERC	Function	Error Description	RAC	Comments
0700	Storage protect key written via an LSSD operation for address 0 and 2 K. Storage protect key is read back via In X'73'. Written and read values are Compared. All 8 possible values are used.	Mismatch during compare.	81A	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields respectively

BD09 - IN X'73' TEST (ADDRESS EXCEPTION KEY)

This routine tests the address exception key in the storage key data register test.

ERC	Function	Error Description	RAC	Comments
0700	Address exception key is written via LSSD operations for address 0 and 2 K. Address exception key is read back via In X'73'. The written and read value are compared. Both 0 and 1 values are used.	Mismatch during compare.	81A	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields respectively

BE01 - IN X'75' TEST

ERC	Function	Error Description	RAC	Comments
0700	Read In X'75' to check for initial value. Set all bits of In X'75' via LLSD operation and read back the contents of In X'75' via indirect operation. Written and read values are compared	Mismatch during compare	878	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields respectively

BE03 - IN X'7D' TEST

ERC	Function	Error Description	RAC	Comments
0700	Read In X'7D' to check for initial value. Set all bits of In X'7D' via LSSD operation after disabling CCU interrupts. Read In X'7D' via indirect operation. Compare written and read values. Reset CCU Hard Check in CCU-to-MOSS Status A register Set CCU Check Reset in diagnostic control register. Enable interrupts	Mismatch during compare	879	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields respectively

BE04 - IN X'7E' TEST

ERC	Function	Error Description	RAC	Comments
0700	Check initial value of In X'7E'. Set on all bits in In X'7E' via an LSSD operation. Read back In X'7E' via indirect operation. Compare written and read values.	Mismatch during compare match	825	Correct and bad value are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Read In X'76'. Check for IOC address exceptions.	No IOC address exception		
0702	Read In X'7D'. Check for POP Parity Error	No POP Parity Error		

BE05 - IN X'7F' TEST

ERC	Function	Error Description	RAC	Comments
0700	Set on bits in In X'7F' via an LSSD operation. Enable CCU Interrupts Read In X'7F' via indirect operation. Compare written and read values.	Mismatch during compare	826	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BE06 - IN X'77' TEST

This routine tests the read latches set for CCU level 2, 3 and 4 interrupt requests in the IO1I register.

ERC	Function	Error Description	RAC	Comments
0700	Initialize the ROSAR, LSAR, MOSS-to-CCU Status reg., and IO1I via LSSD operations. Advance the C-Clock in clock step mode 3 times. Read back the LSSD strings. Compare the data written in IO1I and read in MDOR 3 values are used in IO1I: X'3AAAA' X'35555' X'03131'	Mismatch during compare	879	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BF01 - OUT X'73' ROS CYCLE TEST

This routine checks that during an Out X'73' ROS task, the SAR contains the value of the key and stays unchanged.

ERC	Function	Error Description	RAC	Comments
0700	Write Out X'73' = X'000010'. Read back SAR Compare written and read values.	Mismatch during compare.	81A	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BF02 - OUT X'73' STORAGE PROTECT KEY TEST

ERC	Function	Error Description	RAC	Comments
0700	Write Out X'73' for addr. 0 with all possible values of storage protect key (0-7). Read back SKDR via LSSD operation. Compare written and read values.	Mismatch during compare.	85C	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Do the same for address 2K.		85C	

BF03 - OUT X'73' READ-ONLY KEY TEST

ERC	Function	Error Description	RAC	Comments
0700	Write Out X'73' for addr. 0 with values 1 and 0. Read back SKDR via LSSD operation. Compare written and read values	Mismatch during compare	81A	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Do the same for address 2K.		81A	

BF04 - OUT X'73' ADDRESS EXCEPTION INDICATOR TEST

ERC	Function	Error Description	RAC	Comments
0700	Write Out X'73' for address 0 with values 1 and 0. Read back SKDR via LSSD operation. Compare written and read values.	Mismatch during compare.	81A	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BF05 - OUT X'73' USER KEY TEST

ERC	Function	Error Description	RAC	Comments
0700 to 0714	Write Out X'73' for all user key pointer registers and 5 user key program levels. All values of user key (0-7) are used. Read back user keys register via LSSD operations and compare values.	Mismatch during compare. ERC number gives failing user key pointer register or user key program level: 700 user key pointer reg 0 701 user key pointer reg 1 702 user key pointer reg 2 703 user key pointer reg 3 704 user key pointer reg 4 705 user key pointer reg 5 706 user key pointer reg 6 707 user key pointer reg 7 708 user key pointer reg 8 709 user key pointer reg 9 70A user key pointer reg A 70B user key pointer reg B 70C user key pointer reg C 70D user key pointer reg D 70E user key pointer reg E 70F user key pointer reg F 710 user key program level 1 711 user key program level 2 712 user key program level 3 713 user key program level 4 714 user key program level 5	81A	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BF06 - OUT X'73' MODIFY FUNCTION TEST

ERC	Function	Error Description	RAC	Comments
0700	Write Out X'73' storage key = 0 with modify. Write Out X'73' storage key = 1 without modify. Read In X'73' and test that storage key = 0.	Tested key not equal to 0.	81A	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Write Out X'73' exception key = 0 with modify. Write Out X'73' exception key = 1 without modify. Read In X'73' and test that storage key = 0.	Tested key not equal to 0.	81A	
0702	Write Out X'73' read only key = 0 with modify. Write Out X'73' read only key = 1 without modify. Read In X'73' and test that read only key = 0.	Tested key not equal to 0.	81A	

BG01 - KEY STORAGE BIT TEST

Write and read the whole key storage with patterns X'55 and X'AA'.

1. Using Out X'73', write the Address Exception key, the Read Only key, and the Storage Protect key, for every 9-bit word of key storage.

2. Read back and verify the contents.

ERC	Function	Error Description	RAC	Comments
0701	Write/read all key storage words.	Read data not equal to expected data.	817	ERR Bit field on the screen shows the address of the wrong word

BG02 - KEY STORAGE ADDRESSING TEST

Address every word of the whole key storage.

Write in each word of key storage its own address and read it back via LSSD operations. This is done in both ascending and descending order.

ERC	Function	Error Description	RAC	Comments
0701	Write/read word-by-word the whole key storage.	Read data not equal to expected data.	817	ERR BIT field on the screen shows the address of the wrong word

BH01 - OUT X'7A' TEST (PART 1)

Check the initial values of the high and low resolution timers.

ERC	Function	Error Description	RAC	Comments
0700	Write and read LSSD strings with clock off. Check the value of the high resolution timer.	Value not equal to 0.	812	Correct and bad values are displayed on screen in EXP DATA and RCV DATA fields respectively
0701	Check value of the low resolution timer	Value not equal to 0.	812	

BH02 - OUT X'7A' TEST (PART 2)

Test the incrementation of the high resolution timer.

ERC	Function	Error Description	RAC	Comments
0700	Mask all interrupt levels. Initialize high resolution timer to X'3FFFFF' via LSSD operation with C-Clock off. Step the CCU 16 steps. Read In X'7A' - should be X'00000F'.	In X'7A' not equal to X'0000F'.	87A	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BH03 - OUT X'7A' TEST (PART 3)

Test the incrementation of the low resolution timer.

ERC	Function	Error Description	RAC	Comments
0700	Mask all interrupt levels. Initialize high resolution timer to X'3FFFFF'. Initialize low res. timer to X'000FF8' via LSSD operation with C-Clock off. Step the CCU 16 steps. Read contents of the low resolution timer; should be X'000008'.	Low resolution timer not equal to X'000008'.	812	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Read the contents of high resolution timer; should be X'000000'.	High resolution timer not equal to X'000000'.	812	

BH04 - OUT X'7A' TEST (PART 4)

This routine checks the timer as a utilization counter.

ERC	Function	Error Description	RAC	Comments
0700	Mask all interrupt levels. Initialize high resolution timer to X'3FFFFF'. Initialize low resolution timer to X'000FF8'. Set utilization counter mode Read back In X'7A'. High resolution timer should stay at X'3FFFFF'.	High resolution timer not equal to X'3FFFFF'.	871	Correct and bad values are displayed onthe screen in EXP DATA and RCV DATA fields respectively
0701	Perform Out X'7A' to enable low resolution counter. Read back In X'7A' - should be X'000000'.	Low resolution timer not equal to X'000000'.	871	

BI01 - OUT X'77' TEST

This routine tests the reset of level 1, 2, 3, and 4 interrupt requests.

ERC	Function	Error Description	RAC	Comments
0700	Write Out X'77', with all bits on, then unmask. Read In X'7F' - should be 0 (all interrupts level 2, 3, and 4 reset).	In X'7F' not equal to 0.	87B	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Read In X'7E' - should be 0 (all level 1 interrupts reset).	In X'7E' not equal to 0.	87B	

BI02 - OUT X'79' AND IN X'79' TEST

ERC	Function	Error Description	RAC	Comments
0700	Set enter level 4 latch. Write Out X'79' with X'2308' Read Out X'79' via LSSD and check value.	Out X'79' not equal to X'2308'.	87C	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Read CCU-to-MOSS Status A register; should contain X'50'.	MOSS-to-CCU Status A register not equal to X'50'.	811	
0702	Read In X'79'; should be X'0300'.	In X'79' not equal to X'0300'.	877	
0703	Read CCU-to-MOSS Status F register should contain X'C0'.	CCU-to-MOSS Status F register not equal to X'C0'.	81A	
0704	Write Out X'79' = X'4404'. Read Out X'79' via an LSSD. Should contain X'0300'.	Out X'79' not equal to X'0300'.	87D	
0705	Reset Pgm Reg IPL in CCU-to-MOSS Status A reg. Read this register (should contain X'10').	MOSS-to-CCU Status A reg not equal to X'10'.	811	
0706	Read In X'79'; should contain X'0300'.	In X'79' not equal to X'0300'.	81A	
0707	Read CCU-to-MOSS Status F register; should contain X'C0'.	CCU-to-MOSS Status F register not equal to X'C0'.	81A	
0708	Write 'Set AIO Stop' in Out X'79.	'AIO Stop' bit not raised in Mode Control reg B.	86A	
0709	Write 'Reset AIO Stop' in Out X'79'.	'AIO Stop' bit not reset in Mode Control reg B.	86A	

BI03 - OUT X'7E' AND OUT X'7F' TEST

This routine tests the set and reset of the program interrupt masks.

ERC	Function	Error Description	RAC	Comments
0700	Write Out X'7E' all bits. Read LSSD and check masks for all levels (X'1F').	Masks are not set for all levels.	811	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Write Out X'7F' all bits. Read LSSD and check masks for all levels (X'00')	Masks are not reset	811	

BI04 - CCU/STORAGE SCOPING ROUTINE

This CCU manual intervention routine exercises the CCU to main storage data and control lines. It does not isolate an FRU, but allows looping with selected command, address, and data patterns for scoping.

ERC	Function	Error Description	RAC	Comments
	Error during scoping routine when requested	Prompting messages for CE (see MIM 1, Chapter 10, Section 2).	89F	Reserved for CCU storage scoping

BI05 - NETWORK POWER OFF TEST

This is a manual intervention routine.

ERC	Function	Error Description	RAC	Comments
0700	Check that when the network power off bit is on using Out X'79' (byte 0, bit 4), the 3725 is powered off.	Network power off did not occur.	8FA	

BJ01 - STORAGE FIRST ACCESS TEST

1. Write, then read, the first and second halfwords of the first card in the ECC disable mode.
2. Compare the written and read data.
3. If more than two bits in error, select the card.
4. If all cards have 2 or more bits in error, test byte select.
5. Test for a high level interrupt.

ERC	Function	Error Description	RAC	Comments
0701	Check MTAR contents after a write command.	MTAR does not contain the right address.	877	Correct and bad values are respectively
0702	Check SAR contents after a write command.	SAR does not contain the right address.	87F	displayed on the screen in EXP DATA and RCV
0703	Check MTAR contents after a read command.	MTAR does not contain the right address.	877	DATA fields respectively
0704	Check SAR contents after a read command.	SAR does not contain the right address.	87F	
0705	Writing and/or reading the CCU storage is not possible.	Read value is 'all ones'.	817 780	ERR BIT field equal to X'00F5'
	Try to find an error free storage card.	Displayed card has been found with 2-bit error.	8E1 to 8E8 789 to 78C	ERR BIT field equal to X'00F6'
	Check byte select function.	There is a byte select error within the same halfword Y/X or 0/1).	8EE 781	ERR BIT field equal to X'00F7'
		There is a byte select error within the same word (Y/1, Y/0, X/1, X/0).	81D	
0708	Hard Check condition has been detected after write then read command.	Hard Check is due to 1 of 3 storage related checkers: - 2-bit error - storage control error - storage address/data parity error	 8EB 782 8EC 783 8ED 784	ERR BIT field equal to X'00F8'
	Hard Check conditions have been detected.	Hard Check is due to more than one storage related checker being set.	8EA 785	ERR BIT field equal to X'00F9'
	Address exception checking.	Address exception has been erroneously found on.	838 786	ERR BIT field equal to X'0F10'

BJ02 - ECC 1-BIT ERROR CORRECTION TEST

1. Write and read the halfwords at address 0 and at address 2 in the ECC disable mode.
2. If no error, write the whole word with 1-bit error in the ECC transparent mode.
3. Read the same word in the ECC enable mode, and check that the 1-bit error was corrected.

Note: Three different patterns are used and all ECC and parity bits are exercised.

ERC	Function	Error Description	RAC	Comments
0701	Write/read 2 halfwords at address 0 to 3 in ECC disable mode.	A 2-bit error has been detected in data and/or parity bits:	8E1	
		- within the same halfword		ERR BIT field = X'00F1 '
		- 1-bit error by halfword		ERR BIT field = X'00F0 '
	Simulate a 1-bit error and check that ECC mechanism corrects the error.	A 1-bit error has not been corrected.	883	ERR BIT field = X'00F2'
0703	Check ECC checker.	A 2-bit error has been detected in ECC bits.	883	ERR BIT field is meaningless

BJ03 - ECC TWO-BIT ERROR DETECTION TEST

1. Write and read the full word at address 0 in the ECC disable mode.
2. If no error is detected, write the same pattern with two bits modified in the ECC transparent mode.
3. Enable the ECC and read.
4. Check for 2-bit errors.

Notes:

1. The test uses the pattern X'00' twice, and the pattern X'FF' twice. All ECC and parity bits are exercised.
2. This routine must not be specifically selected.

ERC	Function	Error Description	RAC	Comments
0701	Simulate a 2-bit error and check that the ECC mechanism detects the error.	A 2-bit error has not been detected by ECC mechanism.	884	ERR BIT field is meaningless
0702		After simulation of 2-bit error, MDOR Parity Error occurred instead of 2-bit error. In order to establish if the MDOR Parity Error was due to incorrect 2-bit error handling by the ECC mechanism or due to the fact that an error was forced for a bit which was already in error. After using several data and simulation it is possible to say that the ECC mechanism is not working properly.		

BJ04 - STORAGE TEST PATTERN

1. Run the storage test pattern in the ECC disable mode with data = X'AA'.
2. Wait for a high level interrupt: if there is no high level interrupt within 1.3 seconds, report an error.
3. If an address exception occurs at the end of installed storage, verify the data at both address 0 and at the highest installed address.

ERC	Function	Error Description	RAC	Comments
0700	Check that CCU remains busy during the storage test pattern command execution.	CCU is found not busy immediately after the storage test pattern command is issued.	839	ERR BIT field is meaningless
0701	The storage test pattern command must end with an HLIR due to an address exception or a Hard Check (Hard Check being a possible consequence of MDOR parity error, a 1-bit error, or a 2-bit error).	Neither address exception nor Hard Check due to these 2 specific causes has been detected.	881	ERR BIT field is meaningless
0702	Read data contents of address 0.	This first address does not contain written pattern.	87F	ERR BIT contains the bits found in error
0703	Read data contents of last installed halfword.	This last address does not contain written pattern.	882	ERR BIT contains the bits found in error

BJ21 - CONTROL LINES WRAP TEST

This routine is used to check for permanent errors on the CCU bus storage related control lines.

It is activated only on EC A04404 3725 because it uses the wrap capability of certain CCU storage related control lines onto the storage data bus byte 0, which is available only at this EC level.

ERC	Function	Error Description	RAC	Comments
0700	Issue an Output X'74', then a read storage to fetch the control lines status onto data bus byte 0.	One control line found clamped on.	788	ERR BIT field is equal to X'0001'
		Read value is all ones due to SP write inhibit.	817	ERR BIT field is equal to X'0002'
		Read value is all zeros due to storage grant timeout.	794	ERR BIT field is equal to X'0003'

BJ22 - ADDRESS BUS WRAP TEST

This routine is used to check for permanent errors on the CCU storage address bus. The patterns used are X'2A55CC' and X'15AA33'.

It is activated only on EC A04404 3725 because it uses the wrap capability of each address bus byte onto the storage data bus byte 0, which is available only at this EC level.

ERC	Function	Error Description	RAC	Comments
0700	Issue an Output X'74', then a read storage to fetch each address bus byte onto data bus byte 0.	All patterns are found in error.	787	ERR BIT field is equal to X'0001'
		Some patterns are found in error.	787	ERR BIT field is equal to X'0002'
		Only one pattern is found in error.	78D to 793	ERR BIT field is equal to X'0003'

BJ23 - ECC ONLY MODE FACILITY TEST

This routine is used to check the ECC only mode that is available at EC A04404. It allows access to the CCU storage data bus without involving the CCU storage cards. In the routine, the read and read-modify-write commands are tested together with the CCU storage data bus. The patterns used are X'AAAAAAAA' and X'55555555'.

It is activated only on EC A04404 3725.

ERC	Function	Error Description	RAC	Comments
0700	Issue an Output X'74', then a read-modify-write, then a read command. The data information is saved inside the ECC card.	All patterns are found in error.	796 to 79C	ERR BIT field is equal to X'0001'
		Some patterns are found in error.	79D to 7A3	ERR BIT field is equal to X'0002'

BJ24 - FORCE STORAGE RELATED CHECKERS TEST

This routine is used to check the CCU storage related checkers and the continuity between the SCTL and the CTL2 cards.

It is activated only on EC A04404 3725.

ERC	Function	Error Description	RAC	Comments
	Issue an Output X'74' to force a storage related checkers to 'on'.	The selected checker is not found 'on'.	794	ERR BIT field is meaningless.
0700	Storage address/data parity errors.			
0701	Storage 2-bit error.			
0702	Storage control error.			

BJ25 - NEW ECC MECHANISM TEST

This routine is used to check that the ECC mechanism installed at EC A04404 3725 works correctly, using the following facilities:

- Single hard-bit error correction.
- Single soft-bit error correction.
- Single hard-bit, plus single soft-bit error correction.
- Double hard-bit error detection.
- Double soft-bit error detection.

The soft and hard errors are both simulated using an Output X'74' command.

The routine is activated only on EC A04404 3725.

ERC	Function	Error Description	RAC	Comments
0700	Single hard-bit error correction.	Single hard-bit error really present at the addressed location was not corrected with the ECC enabled.	795	ERR BIT field is equal to X'0001'
		The addressed location being error free, a single hard-bit simulation was done, but not corrected with the ECC enabled.	795	ERR BIT field is equal to X'0002'
		The addressed location had a single soft-bit error. A single hard-bit simulation was done but, not corrected with the ECC enabled.	795	ERR BIT field is equal to X'0003'
		No '+1-bit error corrected' information was given after correction of a simulated single hard-bit error.	795	ERR BIT field is equal to X'0004'
0710	Single soft-bit error correction.	Single soft-bit error was simulated using the ECC transparent mode at an error free location, but is not corrected when the ECC is enabled.	795	ERR BIT field is meaningless.

ERC	Function	Error Description	RAC	Comments
	Double bit error (one soft and one hard).		7A4	ERR BIT field is meaningless.
0720		Only the soft-bit error was simulated, one single hard bit error being detected at the test location. These two bit errors are not corrected when the ECC is enabled.		
0721		A single soft-bit error and a single hard-bit error were simulated, but not corrected when ECC is enabled.		
	Double hard-bit error detection.		7A4	ERR BIT field is meaningless.
0730		Only a single hard-bit error was simulated at the test location, but a real one was detected as well. This two-bit error is not detected when the ECC is enabled.		
0731		A two hard-bit error was simulated, but not detected when the ECC is enabled.		
0740	Double soft-bit error detected.	A two soft-bit error was simulated, but not detected when the ECC is enabled.	7A4	ERR BIT field is meaningless.

BK01 - DATA PATH TEST

Write and read different patterns in the first and second halfwords of storage.

ERC	Function	Error Description	RAC	Comments
0700	Compare written and read data.	Mismatch in the data.	88E	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Test Hard Check.	Hard Check with storage address/data parity occurred.	8F0	

BK02 - SCAN WHOLE STORAGE TEST (2-BIT ERROR DETECTION)

1. Run the storage test pattern in the whole installed storage with 4 different data patterns.
2. Wait for an address exception: if no address exception occurs, then suspect a storage card; execute the following steps.
3. Limit the storage size using the address exception key to 1 card and restart the scan.
4. Continue with the next cards, one-by-one, to find out the card in error.

ERC	Function	Error Description	RAC	Comments
0700	Check that CCU remains busy during the storage test pattern command execution.	CCU is found not busy immediately after the storage test pattern is issued.	839	ERR BIT field is meaningless
0701	Test each storage card in turn to isolate the bad one.	Storage address/data parity error detected on first card	81E	ERR BIT field is meaningless
		One storage card is found in error.	8E1 to 8E8 789 to 78C	ERR BIT field is meaningless

BK03 - STORAGE ADDRESSING TEST

1. Test the byte select mechanism - write a specific byte in fullword 0 and check after reading it back.
2. Test the word addressing mechanism - address all 128 fullwords in the first block of storage in both ascending and descending order.
3. Test the refresh mechanism - address the 1st fullword of each block of 512 bytes in both ascending and descending order.
4. Write the fullword at address X'10000'.
5. Write the first fullword of each installed storage card.

ERC	Function	Error Description	RAC	Comments
0700	Write and read specific byte at address 0.	Read data not equal to expected data.	885	With the exception of ERC 707 correct and and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Address 128 fullwords in ascending mode.	Read data not equal to expected data.		
0702	Address 128 fullwords in descending mode.	Read data not equal to expected data.		
0703	Address 1st fullword of each 512-byte block in ascending mode.	Read data not equal to expected data.		
0704	Address 1st fullword of each 512-byte block in descending mode.	Read data not equal to expected data.		
0705	Write a fullword at address X'10000'.	Read data not equal to expected data.		
0706	Write fullword 0 of a specific card and check data	Read data not equal to expected data.		ERR BIT field = X'00F6'
	Check that all others cards stay at initial value X'00'.	Non zero value found at address 0 of a non-selected storage card.		ERR BIT field = X'00F7'.

BK04 - STORAGE TEST (16K WORDS)

1. Write the storage, halfword by halfword, from 0 to 16K.
2. Read each halfword and verify the contents.

ERC	Function	Error Description	RAC	Comments
0700	Write and read a halfword in the range 0 - 16K.	Read data not equal to expected data.	885	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BK05 - INITIAL IPF CONTROL MECHANISM TEST

1. Write storage position 0 with the fullword X'AAAA5555'.
2. Initialize the IAR to X'00'.
3. Initialize the IPF (instruction pre-fetch) function.
4. Read POPA and POPB.

ERC	Function	Error Description	RAC	Comments
0701	Read POPA.	POPA does not contain X'AAAA'.	819	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0702	Read POPB.	POPB does not contain X'5555'.	819	

BL01 - WRITE CHARACTER - NO INCREMENT, NO LOOP

1. Write the character X'AA' (ROS command) at address 0.
2. Verify the contents of the SAR for a correct address.
3. Read the character and verify that there is no error.

ERC	Function	Error Description	RAC	Comments
0701	Verify the address in SAR.	SAR address incorrect (should be X'00').	87F	Correct and bad values are displayed on the screen in EXP DATA AND RCV DATA fields respectively
0702	Verify read data (byte 0).	Data not correct.	8FF	

BL02 - WRITE HALFWORD - NO INCREMENT, LOOP

1. Write the halfword X'AAAA' (ROS command) at address 0.
2. Verify that the busy bit is on in the CCU-to-MOSS Status B register.
3. Wait for 100 milliseconds and reset the busy bit in order to stop the loop.
4. Verify the SAR for the correct address.
5. Verify the MTAR for the correct address.
6. Read the data and verify.

ERC	Function	Error Description	RAC	Comments
0700	Test busy bit.	No busy bit.	839	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Verify the storage address in SAR.	Bad storage address.	819	
0702	Verify the storage address in MTAR.	Bad storage address.	819	
0703	Verify read data.	Read data not equal to expected data.	819	

BL03 - READ HALFWORD NO INCREMENT, LOOP

1. Write the halfword X'AAAA' at address 0.
2. Start read halfword (ROS command).
3. Verify that the busy bit is on in the CCU-to-MOSS Status B register.
4. Loop for 100 milliseconds and reset the busy bit in order to stop the loop.
5. Verify the SAR for a correct address.
6. Verify MTAR for a correct address.

ERC	Function	Error Description	RAC	Comments
0700	Test busy bit.	No busy bit.	839	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Verify the storage address in the SAR.	Bad storage address.	819	
0702	Verify the storage address in MTAR.	Bad storage address.	819	

BL04 - WRITE CHARACTER NO INCREMENT, LOOP

1. Write the character at address 0 with a ROS command.
2. Verify that the busy bit is on in the CCU-to-MOSS Status B register.
3. Loop for 100 milliseconds and reset the busy bit in order to stop the write loop.
4. Verify the SAR for a correct address.
5. Verify the written data.

ERC	Function	Error Description	RAC	Comments
0700	Test busy bit.	No busy bit.	839	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Verify the storage address in SAR.	Bad storage address.	87F	
0702	Verify read data.	Read data not equal to expected data.	8FF	

BL05 - SINGLE ADDRESS TEST PATTERN

1. Start the test pattern X'AAAA' at address 0 using a ROS command.
2. Verify that the busy bit is on in the CCU-to-MOSS Status B register.
3. Scan for 100 milliseconds and reset the busy bit in order to stop the scan.
4. Verify the SAR for a correct address.
5. Verify the data in the MTDR.
6. Read the data (via an indirect operation) and verify.

ERC	Function	Error Description	RAC	Comments
0700	Test busy bit.	No busy bit.	839	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Verify the storage address in SAR.	Bad storage address.	87F	
0702	Verify the data in MTDR.	Wrong data in MTDR.	877	
0703	Read data by the indirect operation.	Wrong data read.	880	

BL06 - SINGLE ADDRESS SCAN TEST

1. Write address 0 with data X'AAAA'.
2. Start a scan at address 0 using a ROS command.
3. Test that the busy bit is on in the CCU-to-MOSS Status B register.
4. Scan for 100 milliseconds and reset the busy bit in order to stop the scan.
5. Verify the storage address in the SAR.
6. Verify the data in the MTDR.

ERC	Function	Error Description	RAC	Comments
0700	Test busy bit	No busy bit	839	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Verify the storage address in SAR	Bad storage address	819	
0702	Verify the data in MTDR	Wrong data in MTDR	819	

BL07 - WRITE HALFWORD, INCREMENT, NO LOOP

1. Write X'AAAA' at address 0 with ROS command.
2. Read the address in the MTAR and verify that the address was incremented by 2.
3. Verify the written data.

ERC	Function	Error Description	RAC	Comments
0701	Verify address incrementation in MTAR	Wrong address in MTAR	819	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0702	Verify written data	Read data not equal to expected data.	819	

BL08 - READ HALFWORD, INCREMENT, NO LOOP

1. Write the halfword X'AAAA' at address 0.
2. Read the data at address 0 using a ROS command.
3. Read the address in the MTAR and verify that the address was incremented by 2.
4. Verify the read data.

ERC	Function	Error Description	RAC	Comments
0701	Verify address incrementation in MTAR	Wrong address in MTAR	819	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0702	Verify read data	Wrong data read	819	

BL09 - WRITE HALFWORD, INCREMENT, LOOP

1. Write the halfword X'AAAA' starting at address 0 using a ROS command.
2. Test that the busy bit is on in the CCU-to-MOSS Status B register.
3. Continue the scan for 100 milliseconds and reset the busy bit in order to stop the scan.

ERC	Function	Error Description	RAC	Comments
0700	Test busy bit	No busy bit	839	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Test data at address 0	Wrong data at address 0	819	
0702	Test data at the highest installed address	Wrong data at the highest installed address	819	
0703	Verify address in MTAR	Address = 0 (no loop)	819	

BL0A - READ HALFWORD, INCREMENT, LOOP

1. Using write halfword (increment, loop), write the whole storage with the pattern X'AAAA'.
2. Test that the busy bit is on in the CCU-to-MOSS Status B register.
3. Wait for 1.3 seconds, then reset busy bit in order to stop the write command.
4. Start a Read halfword (increment, loop) command.
5. Test that the busy bit is on in the CCU-to-MOSS Status B register.
6. Wait for 100 milliseconds and reset the busy bit in order to stop the read command.
7. Test the storage address in the MTAR.

ERC	Function	Error Description	RAC	Comments
0700	Test busy bit after write command	No busy bit	839	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0701	Test busy bit after read command	No busy bit	839	
0702	Test storage address value	Address 0 in MTAR	819	

BL0B - STORAGE SCAN TEST

1. Start the storage test pattern at address 0 (write whole storage) with the pattern X'AAAA'.
2. Test that the busy bit is on in the CCU-to-MOSS Status B register.
3. Wait for 1.3 second and reset the busy bit in order to stop the storage test pattern.
4. Reset the high level interrupt flag in the interrupt handler (address exception expected).
5. Start a storage scan command (read whole storage).
6. Test that the busy bit is on in the CCU-to-MOSS Status B register.
7. Wait for 1.3 second and reset the busy bit in order to stop the storage scan.
8. Verify the expected high level interrupt in the interrupt handler.

ERC	Function	Error Description	RAC	Comments
0700	Test busy bit during storage test pattern	No busy bit	839	Correct and bad values are displayed in the screen in EXP DATA and RCV DATA fields respectively
0701	Test busy bit during storage scan	No busy bit	839	
0702	Test high level interrupt (address exception)	No address exception	819	

BM01 - ELEMENTARY XFER

This routine tests the following elementary transfers:

- From the IO1D to the WSDR.
- From the IO1A to the SAR.

ERC	Function	Error Description	RAC	Comments
	IO1D is loaded via an LSSD operation with 3 different values: X'3AAAA' X'35555' X'03131' IO1A is loaded via an LSSD operation with 3 different values: X'070AAAAA' X'07355555' X'00313131' Gates are initialized (CTI1, IOCB). Do one clock step.			Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0700	Verify that WSDR contains data from IO1D	Data not transferred correctly		
0701	Verify that SAR contains data from IO1A	Data not transferred correctly	8DA 8DB 8DC 8DD 8DE 8DF 8E0	

BM02 - ELEMENTARY TRANSFER

This routine tests the following elementary transfers:

- From the WSDR to the IO1D.
- From the SAR to the IO1A.

ERC	Function	Error Description	RAC	Comments
	WSDR is loaded via an LSSD operation with 3 different values: X'0300AAAA' X'03005555' X'00003131' SAR is loaded via an LSSD operation with 3 different values: X'070AAAAA' X'07355555' X'00043131' Initialize following gates: RPTY, STUI, RSOE, CCUI, RSTG, RZBS, CTI1, CTI3. Do one clock step.			Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively
0700	Verify that IO1A contains data from SAR	Data not transferred correctly		
0701	Verify that IO1D contains data from WSDR	Data not transferred correctly	8DA 8DB 8DC 8DD 8DE 8DF 8E0	

BM03 - IOC STATE STEP TEST VIA LSSD (AIO)

An AIO operation is divided into 22 logical states. Each state is tested with the CCU in clock step mode by reading registers IO1A, IO1D, CTI3, CTI1, and CTI4.

Each ERC corresponds to the test of a particular state.

ERC	Function	Error Description	RAC	Comments
0700 to 0715	Test state of execution of AIO operation	An error in the step	818 8D5 8C5 8D6 8CE 845 8CB 8CC 8D8 8CD 8D9	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

BN01 - IOC STATE STEP TEST VIA LSSD (PIO)

A PIO operation is divided into 13 logical states. Each state is tested with the CCU in clock step mode by reading registers IO1A, IO1D, CTI1, CTI3, and CTI4.

Each ERC corresponds to the test of a particular state.

ERC	Function	Error Description	RAC	Comments
0700 to 070C	Test state of execution of PIO operation	An error in the step	818 85A 81A 8C5 820 8CE 8CE	Correct and bad values are displayed on the screen in EXP DATA and RCV DATA fields respectively

B001 - STORAGE SOLID 1-BIT ERROR DETECTION (MANUALLY INVOKED ROUTINE)

The purpose of this routine is to provide a storage card list for which the number of solid single-bit errors is higher than a given threshold.

A maximum of six cards is called when six or more cards having more 1-bit errors than the fixed threshold are detected.

ERC	Function	Error Description	RAC	Comments
		Prompting messages for CE (see MIM 1, Chapter 2, Section 2, "Extended Troubleshooting Using BERs").		

CA01 - LRI INSTRUCTION AND INSTRUCTION STEP TESTS

1. Verify the loading of the correct byte in the local store and in the corresponding Work register.
2. Verify the setting of the CZ latches.
3. Test the ROS address decode from each POP.
4. Verify the instruction step mode.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in register to prepare the instruction execution.	Register setting is incorrect: 0401 Mode Control Reg B 0402 POPA 0403 Work register 0404 IAR	8A0	Instructions loaded at X'800'
0601 to 0607	LRI 1(0), X'FF' LRI 3(0), X'00' LRI 5(1), X'FF' LRI 7(1), X'00' LRI 3(0), X'00'	After executing LRI instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 Op register 0603 Work register 0604 Local store 0605 CZ latches 0606 IAR 0607 LAR	8A0	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CA02 - INSTRUCTION PRE-FETCH TEST

This routine checks the instruction pre-fetch operation, using the LRI instruction (previously tested by routine CA01).

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Instruction prefetch is initialized with address of the 1st instruction. CCU is stepped and verification is done that correct POP is loaded with correct instruction	After CCU stepping one of registers contains incorrect value: 0401 Mode Control Reg B 0402 POP 0403 IAR 0404 SAR	8A1	Instructions loaded at X'800'
0601 to 0603	LRI instruction set as in CA01	After executing the LRI instruction one of registers contains incorrect value: 0601 Mode Control Reg B 0602 POP 0603 OPRG	8A1	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CA03 - BRANCH INSTRUCTION TEST

1. Check the Op register to verify execution of the correct instruction.
2. Verify the correct updating of the IAR and the SAR.
3. Verify IPF select after the branch.
4. Branch instructions with positive and negative displacement are tested.
5. The CZ latches do not change.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 CZ latches	8A2	Instructions loaded at X'800'
0601 to 0607	B + 002 B - 002 B - 7FE B + 7FE	After executing the B instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 Op reg 0603 IAR 0604 SAR 0605 IPFC 0606 CZ latches 0607 LAR	8A2	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CB04 - BZL INSTRUCTION TEST

1. With the Z latch on, verify the Op register for a correct instruction.
2. Verify SAR, IAR and LAR updating.
3. Verify the value of the CZ latches.
4. With the Z latch off, check for no branch.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 CZ latches	8A2	Instructions loaded at X'800'
0601 to 0607	BZL + 002 BZL - 7FE BZL + 7FE BZL - 002	After executing the BZL instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 OPRG 0603 IAR 0604 SAR 0605 IPFC 0606 CZ latches 0607 LAR	8A2	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CB05 - BCL INSTRUCTION TEST

1. With the C latch on, verify the Op reg for correct instruction.
2. Verify SAR, IAR, and LAR updating.
3. Verify the value of the CZ latches.
4. With the C latch off, check for no branch.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 CZ latches	8A2	Instructions loaded at X'800'
0601 to 0607	BCL + 002 BCL - 7FE BCL + 7FE BCL - 002	After executing the BCL instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 OPRG 0603 IAR 0604 SAR 0605 IPFC 0606 CZ 0607 LAR	8A2	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CB06 - BB INSTRUCTION TEST

1. With the tested bit on, verify the Op reg for correct instruction.
2. Verify SAR, IAR, and LAR updating.
3. Verify the value of CZ latches and check that the local store was not changed.
4. With the tested bit off, check for no branch.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 CZ 0405 LS	8A2	Instructions loaded at X'800'
0601 to 0607	R N M Display BB 1 0 000 + 02 BB 3 1 001 - 7E BB 5 0 110 + 7E BB 7 1 111 - 02 BB 1 1 010 + 02 BB 3 0 011 - 7E BB 5 1 100 + 7E BB 7 0 101 - 02	After executing the BB instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 OPRG 0603 IAR 0604 SAR 0605 IPFC 0606 LS 0607 CZ latches 0608 LAR	8A2	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CC07 - ARI INSTRUCTION TEST

1. Check the Op register for correct instruction.
2. Check the Work register, the Local Store and the CZ latches for the correct result.
3. Check IAR and LAR updating.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work register	816	Instructions loaded at X'800'
0601 to 0607	R N Immed field ARI 1 0 00 ARI 3 1 01 ARI 5 1 FF ARI 7 0 FF	After executing the ARI instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 OPRG 0603 Work register 0604 LS corresponding to Work register 0605 CZ latches 0606 IAR 0607 LAR	816	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CC08 - ORI INSTRUCTION TEST

1. Check the Op register for the correct instruction.
2. Check the Work register, Local Store and CZ latches for result.
3. Check IAR and LAR updating.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg	816	Instructions loaded at X'800'
0601 to 0607	R N Immed field ORI 1 0 00 ORI 3 1 55 ORI 5 1 00 ORI 7 0 AA	After executing the ORI instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 OPRG 0603 Work reg 0604 LS corresponding to Work reg 0605 CZ latches 0606 IAR 0607 LAR	816	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CC09 - XRI INSTRUCTION TEST

1. Check the Op register for the correct instruction.
2. Check the Work register, Local Store and CZ latches for result.
3. Check IAR and LAR updating.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg	816	Instructions loaded at X'800'
0601 to 0607	R N Immed field XRI 1 1 55 XRI 3 0 AA XRI 5 0 00 XRI 7 1 FF	After executing the XRI instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 OPRG 0603 Work reg 0604 LS corresponding to work reg 0605 CZ latches 0606 IAR 0607 LAR	816	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CD0A - NRI INSTRUCTION TEST

1. Check the Op register for the correct instruction.
2. Check the Work register, Local Store and CZ latches for result.
3. Check IAR and LAR updating.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg	816	Instructions loaded at X'800'
0601 to 0607	R N Immed field NRI 1 0 AA NRI 3 1 55 NRI 5 0 55 NRI 7 1 AA	After executing the NRI instruction, one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 OPRG 0603 Work reg 0604 LS corresponding to work reg 0605 CZ latches 0606 IAR 0607 LAR	816	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CD0B - TRM INSTRUCTION TEST

1. Check the Op register for the correct instruction.
2. Check the updating of the CZ latches.
3. Check that the Work register and Local Store are not modified.
4. Check IAR and LAR updating.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg	816	Instructions loaded at X'800'
0601 to 0607	R N Tested bit TRM 1 1 01 TRM 3 0 02 Marked TRM 5 1 40 bit on TRM 7 0 80 TRM 1 0 10 TRM 3 1 20 Marked TRM 5 0 04 bit off TRM 7 1 08	After executing the TRM instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 OPRG 0603 Work reg 0604 LS corresponding to work reg 0605 CZ latches 0606 IAR 0607 LAR	816	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CE0C - XR INSTRUCTION TEST

1. When the Work register R1 is not equal to 0:

   - Check the Op reg for the correct instruction.
   - Check the Work register, Local Store, and CZ latches updating.
   - Check the IAR and LAR updating.

2. When the Work register R1 is not equal to 0:

   - Check that the branch occurred.

ERC	Function	Error Description	RAC	Comments
0401 to 0405	Set values in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg R1 0405 Work reg R2	816	Instructions loaded at X'800'
0601 to 060B	R1 R2 XR 1 0 XR 2 1 XR 3 2 XR 4 3 XR 5 4 XR 6 5 XR 7 6 XR 0 7 XR 0 0	After executing the XR instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 Op reg 0603 IAR 0604 SAR 0605 IPFC 0606 Work reg R1 0607 LS corresponding to work reg R1 0608 Work reg R2 0609 CZ latches 060A IAR 060B LAR	816	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CF0D - LH INSTRUCTION TEST

1. When the Work register R1 is not equal to 0:
   - Check the Op reg for the correct instruction.
   - Check the Work register, Local Store, and CZ latches updating.
   - Check the IAR and LAR updating.
2. When the Work register R1 is equal to 0:
   - Check that the branch occurred.

ERC	Function	Error Description	RAC	Comments
0401 to 0406	Set value in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg (Base reg) 405 LS45 406 Main by address '700'	87F	Instructions loaded at X'800'
0601 to 060A	R B Display LH 2 1 + 7E LH 6 0 + 00 LH 0 5 + 02 LH 0 0 + 04	After executing the LH instruction, one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 OPRG 0603 IAR 0604 SAR 0605 IPFC 0606 Work reg R 0607 LS corresponding to work register R1 0608 CZ latches 0609 IAR 060A LAR	87F	Correct and bad values displayed on the screen in EXP and RDV DATA fields respectively

CF0E - STH INSTRUCTION TEST

1. When base register is not equal to 0, and R is not equal to 0:

   a. Check that storage address = work register B plus displacement.
   b. Check that the data = contents of R.

2. When base register is not equal to 0, and R is equal to 0:

   a. Check that storage address = work register B plus displacement.
   b. Check that data = Local Storage X'68'

3. When base register is equal to 0, and R is not equal to 0:

   a. Check that storage address = Local Storage X'45' plus displacement.
   b. Check that the data = contents of R.

4. When base register is equal to 0, and R is equal to 0:

   a. Check that storage address = Local Storage X'45' plus displacement.
   b. Check that data = Local Storage X68'

5. Check the Op register for the correct instruction.

6. Check that the CZ latches do not change.

7. Check IAR and LAR updating.

ERC	Function	Error Description	RAC	Comments
0401 to 0407	Set value in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg (Base reg) 0405 LS corresponding to R 0406 LS45 0407 LS68	87F	Instructions loaded at X'800'
0601 to 0606	R B Display STH 2 1 + 7E STH 0 5 + 00 STH 6 0 + 02 STH 0 0 + 04	After executing the STH instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 Op reg 0603 Storage addr X'700' 0604 CZ latches 0605 IAR 0606 LAR	87F	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CG0F - ST INSTRUCTION TEST

1. When base register is not equal to 0, and R is not equal to 0:

   a. Check that storage address = work register B plus displacement.
   b. Check that the data = contents of R.

2. When base register is not equal to 0, and R is equal to 0:

   a. Check that storage address = work register B plus displacement.
   b. Check that data = Local Storage X'68'

3. When base register is equal to 0, and R is not equal to 0:

   a. Check that storage address = Local Storage X'45' plus displacement.
   b. Check that the data = contents of R.

4. When base register is equal to 0, and R is equal to 0:

   a. Check that storage address = Local Storage X'45' plus displacement.
   b. Check that data = Local Storage X68'

5. Check the Op register for the correct instruction.

6. Check that the CZ latches do not change.

7. Check IAR and LAR updating.

ERC	Function	Error Description	RAC	Comments
0401 to 0407	Set value in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg R 0405 LS corresponding to R 0406 LS46 0407 LS68	87F	Instructions loaded at X'800'
0601 to 0606	R B Display ST 5 1 + 7C ST 0 3 + 00 ST 7 0 + 04 ST 0 0 + 08	After executing the ST instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 Op reg 0603 Storage addr X'700' 0604 CZ latches 0605 IAR 0606 LAR	87F	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CG10 - INPUT INSTRUCTION TEST

1. When the Work register R is not equal to 0:
   - Check the Op reg for the correct instruction.
   - Check the Work register and Local Store for correct result.
   - Check that the CZ latches are not modified.
   - Check the IAR and LAR updating.
2. When the Work register R is equal to 0:
   - Check that the branch occurred.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set value in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 LS corresponding to Input	819	Instructions loaded at X'800'
0601 to 060A	R E IN 1 00 IN 2 20 IN 0 23 IN 0 20 IN 4 7D IN 0 7F IN 5 70 IN 0 70 IN 6 74 IN 0 74	After executing the Input instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 Op reg 0603 IAR 0604 SAR 0605 IPFC 0606 Work register R1 0607 Local Store 0608 CZ latches 0609 IAR 060A LAR	819	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CH11 - OUTPUT INSTRUCTION TEST

1. When register E is not equal to 0:
   - Check the Op reg for the correct instruction.
   - Check the Work register and Local Store for correct result.
   - Check that the CZ latches are not modified.
   - Check the IAR and LAR updating.
2. When register E is equal to 0:
   - Check that the branch occurred.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set value in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg R	819	Instructions loaded at X'800'
0601 to 060A	R E ----|--|---| OUT 1 00 OUT 0 5F OUT 2 23 OUT 0 24 OUT 5 20 OUT 0 20	After executing the Output instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 Op reg 0603 IAR 0604 SAR 0605 IPFC 0606 Work register 0607 Local Store 0608 CZ latches 0609 IAR 060A LAR	819	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

CH12 - OUT X'70', X'71', AND X'72' TEST

1. Out X'70'.
   - Check that the Op register contains the correct instruction.
   - Check that a HLIR interrupt was received by the MOSS.
   - Check that the CZ latches have not been modified.
   - Check that the IAR and LAR have been updated correctly.
2. Out X'71' and X'72'.
   - Same as for Out X'70', but check that a LLIR interrupt was received by the MOSS instead of the HLIR.

ERC	Function	Error Description	RAC	Comments
0401 to 0404	Set value in registers to prepare instruction execution	Register setting is incorrect: 0401 Mode Control Reg B 0402 IAR 0403 SAR 0404 Work reg R	819	Instructions loaded at X'800'
0601 to 0607	R E OUT 6 70 OUT 0 70 OUT 7 71 OUT 0 72	After executing the Out instruction one of the tested registers contains an incorrect value: 0601 Mode Control Reg B 0602 Op reg 0603 HLIR/ LLIR 0604 Local Store 0605 CZ latches 0606 IAR 0607 LAR	819	Correct and bad values displayed on the screen in EXP and RCV DATA fields respectively

DA01 - INITIALIZE GENERAL PURPOSE REGISTERS NOT USED BY LEVEL 1

This routine initializes the General Purpose registers not used while running level 1 in the first part of this section. This test is mandatory in order to run the General Purpose register interaction routine (DB61).

ERC	Function	REG	RAC
	Load Output instruction in each General Purpose register		

DB02 - INITIALIZE GENERAL PURPOSE REGISTERS NOT USED BY LEVEL 2

This routine initializes the General Purpose registers not used while running level 2 in the first part of this section. This test is mandatory in order to run the General Purpose register interaction routine (DB67).

ERC	Function	REG	RAC
	Load Output instruction in each General Purpose register		
0001	I/O decode failed		87F

DC03 - INITIALIZE GENERAL PURPOSE REGISTERS NOT USED BY LEVEL 3

This routine initializes the General Purpose registers not used while running level 3 in the first part of this section. This test is mandatory in order to run the General Purpose register interaction routine (DC6F).

ERC	Function	REG	RAC
	Load Output instruction in each General Purpose register		
0001	I/O decode failed		87F

DD04 - INITIALIZE GENERAL PURPOSE REGISTERS NOT USED BY LEVEL 4

This routine initializes the General Purpose registers not used while running level 4 in the first part of this section. This test is mandatory in order to run the General Purpose register interaction routine (DD76).

ERC	Function	REG	RAC
	Load Output instruction in each General Purpose register		
0001	I/O decode failed		87F

DX10 - B INSTRUCTION TEST

This routine checks that the branch instruction is effective and does not alter the CZ latches.

ERC	Function	REG	RAC
	1- Set CZ latches = 01 by loading R1 with zeros	R1(1)	87F
0001	2- Branch with a displacement of 2		
0002	3- Verify that the branch did not alter CZ latches		
	4- Set CZ latches = 10 by loading R1 with ones	R1(1)	
0003	5- Same as 2		
0004	6- Same as 2		

DX11 - LRI, BZL AND BB INSTRUCTION TEST

This routine checks for correct instruction decoding, correct action on the CZ latches, and that the branch is effective.

ERC	Function	REG	RAC
	1- LRI (pattern = X'05') is performed and XORed with same pattern	R1(1)	87F
0001	2- Test that Z latch = 0		
0002	3- LRI (pattern = X'FF') is performed and test CZ = 10	R1(1)	
0003	4- Series of eight BB are performed on R1(1) = X'FF'		
0004	5- Verify that the BB instruction did not alter CZ latches		
0005	6- LRI (pattern = '00') is performed and test CZ = 10	R1(0)	
0006	7- Series of eight BB are performed on R1(0) = '00' and did not alter CZ latches	R1(0)	
0007	8- Previous BB failed		
0008	9- Same as 3 with R1(0)	R1(0)	
0009	10- Same as 4 with byte 0	R1(0)	
000A	11- Same as 5 but branch with absolute value	R1(0)	
000B	12- Same as 6 with byte 1	R1(1)	
000C	13- Same as 7 with byte 1		
000D	14- Same as 8		

DX12 - XRI INSTRUCTION

This routine checks for correct instruction decoding, correct action on the CZ latches, and that the branch on bit is effective.

ERC	Function	REG	RAC
	1- Set R1(1) = X'09' and XORed with pattern = X'05'	R1(1)	87F
	2- XORed with pattern = X'0C'		
0001	3- Verify Z latch = 01		
	4- Set R1 = 'FF00' and perform XRI with pattern = X'FF'		
0002	5- Verify CZ latches		
0003	6- A series of eight BB instruction are performed to see if XRI set wrong bit (byte 1)		
	7- XRI decode using pattern = X'FF'		
0004	8- Same as 5		
0005	9- Same as 6 but R1= X'FF00'		
	10- Same as 7 pattern = X'00'		
0006	11- Same as 5		
0007	12- Same as 6 byte 0 = X'FF'		
	13- Same as 7 pattern = X'00'		
0008	14- Same as 5		
	15- Same as 6 byte 1 = X'00'		
0009	16- XRI set wrong bit		

DX13 - ARI INSTRUCTION

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Add pattern X'05' to pattern X'09' and XOR with pattern X'0E' R1(	) 87	
0001	2- Test Z latch = 0		
	3- Add pattern = X'00' to R1 = X'FF00'	R1(0)	
0002	4- Test CZ latches and byte 0 XORed with pattern = X'FF'		
0003	5- Previous branch on CZ latches failed		
	6- Add pattern = X'00' to R1 = X'0000'	R1(1)	
0004	7- Test CZ latches		
0005	8- XOR R1(1) with pattern X'00' then test Z latch		
	9- Add pattern = X'FF' to R1 = X'0000'		
0006	10- Same as 7		
0007	11- Same as 8 with pattern = X'FF'		
	12- Add pattern = X'FF' to R1 = X'FF00'		
	13- Add pattern = X'FF' to R1 = X'FFFF'		
0008	14- Test CZ latches = 10		
	15- XOR R1(1) with pattern = X'FE'		
0009	16- Verify Z latch		

DX15 - DATA FLOW PATH BYTE ONE (ZEROS PATTERN)

This routine makes successive tests with the Branch On Bit Instruction.

ERC	Function	REG	RAC
	1- Set R1(1) = X'01' only bit 7 = 1	R1(1)	87F
0001	2- Test CZ latches		
0002	3- Perform BB instruction to test zeros pattern		
0003	4- Set R1(1) = X'00' by XRI and test Z latch		
	5- Set R1(1) = X'02' only bit 6 = 1		
0004	6- Same as 2		
0005	7- Same as 3		
0006	8- Same as 4		
	9- Set R1(1) = X'04' only bit 5 = 1		
0007	10- Same as 2		
0008	11- Same as 3		
0009	12- Same as 4		
	13- Set R1(1) = X'08' only bit 4 = 1		
000A	14- Same as 2		
000B	15- Same as 3		
000C	16- Same as 4		
	17- Set R1(1) = X'10' only bit 3 = 1		
000D	18- Same as 2		
000E	19- Same as 3		
000F	20- Same as 4		
	21- Set R1(1) = X'20' only bit 2 = 1		
0010	22- Same as 2		
0011	23- Same as 3		
0012	24- Same as 4		
	25- Set R1(1) = X'40' only bit 1 = 1		
0013	26- Same as 2		
0014	27- Same as 3		
0015	28- Same as 4		
	29- Set R1(1) = X'80' only bit 0 = 1		
0016	30- Same as 2		
0017	31- Same as 3		
0018	32- Same as 4		
	33- Set R1(1) = X'AA' bits 0, 2, 4, and 6 = 1		
0019	34- Same as 2		
001A	35- Perform BB and B instructions to test alternate bits		
001B	36- Same as 4		

DX16 - DATA FLOW PATH BYTE ONE (ONES PATTERN)

This routine makes successive tests with the Branch On Bit Instruction.

ERC	Function	REG	RAC
	1- Set R1(1) = X'FE' only bit 7 = 0	R1(1)	87F
0001	2- Test CZ latches		
0002	3- Perform BB and B instruction to test ones pattern		
0003	4- Set R1(1) = X'00' by XRI and test Z latch		
	5- Set R1(1) = X'FD' only bit 6 = 0		
0004	6- Same as 2		
0005	7- Same as 3		
0006	8- Same as 4		
	9- Set R1(1) = X'FB' only bit 5 = 0		
0007	10- Same as 2		
0008	11- Same as 3		
0009	12- Same as 4		
	13- Set R1(1) = X'F7' only bit 4 = 0		
000A	14- Same as 2		
000B	15- Same as 3		
000C	16- Same as 4		

DX18 - DATA FLOW PATH BYTE ONE (ONES PATTERN)

This routine makes successive tests with the Branch On Bit Instruction.

ERC	Function	REG	RAC
	1- Set R1(0) = X'EF' only bit 3 = 0	R1(0)	87F
0001	2- Test CZ latches		
0002	3- Perfor BB and B instructions to test ones pattern		
0003	4- Set R1(0) = X'00' by XRI and test Z latch		
	5- Set R1(0) = X'DF' only bit 2 = 0		
0004	6- Same as 2		
0005	7- Same as 3		
0006	8- Same as 4		
	9- Set R1(0) = X'BF' only bit 1 = 0		
0007	10- Same as 2		
0008	11- Same as 3		
0009	12- Same as 4		
	13- Set R1(0) = X'7F' only bit 0 = 0		
000A	14- Same as 2		
000B	15- Same as 3		
000C	16- Same as 4		

DX19 - DATA FLOW PATH BYTE ZEROS (ZEROS PATTERN)

This routine makes successive tests with the Branch On Bit Instruction.

ERC	Function	REG	RAC
	1- Set R1(0) = X'01' only bit 7 = 1	R1(0)	87F
0001	2- Test CZ latches		
0002	3- Perform BB and B instructions to test zeros pattern		
0003	4- Set R1(0) = X'00' by XRI and test Z latch		
	5- Set R1(0) = X'02' only bit 6 = 1		
0004	6- Same as 2		
0005	7- Same as 3		
0006	8- Same as 4		
	9- Set R1(0) = X'04' only bit 5 = 1		
0007	10- Same as 2		
0008	11- Same as 3		
0009	12- Same as 4		
	13- Set R1(0) = X'08' only bit 4 = 1		
000A	14- Same as 2		
000B	15- Same as 3		
000C	16- Same as 4		
	17- Set R1(0) = X'55' with bits 1, 3, 5, and 7 = 1		
000D	18- Same as 2		
000E	19- Perform BB and B instruction to test alternate bits		
000F	20- Same as 4		

DX1B - ORI INSTRUCTION TEST

This routine checks for correct instruction decoding and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Set R1(1) = X'09' and OR with pattern = X'05'	R1(1)	
0001	2- XOR with pattern = X'0D' and verify Z latch		87F
	3- Set R1(1) = X'00' and OR with pattern = X'FF'		
0002	4- Test CZ latches		
0003	5- Same as 2 with pattern = X'FF'		
	6- Set R1 = X'FF00' and OR R1(1) with pattern = X'00'		
0004	7- Same as 4		
0005	8- Same as 2 with pattern = X'00'		
	9- OR with pattern = X'00'	R1(0)	
0006	10- Same as 4		
0007	11- Same as 2 with pattern = X'FF'		
	12- OR R1(0) twice with pattern = X'FF'		
0008	13- Same as 4		
0009	14- Same as 2 with pattern = X'FF'		

DX1C - NRI INSTRUCTION TEST

ERC	Function	REG	RAC
	1- Set R1(1) = X'09' and NOR with pattern = X'05'	R1(1)	
0001	2- XOR with pattern = X'01' and verify Z latch		87F
	3- Set R1 = X'FF00' and NOR with pattern = X'00'		
0002	4- Test CZ latches		
0003	5- Same as 2 with pattern = X'00'		
	6- NOR with pattern = X'FF'		
0004	7- Same as 4		
0005	8- Same as 2 with pattern = X'FF'	R1(0)	
	9- XOR with pattern = X'FF' and NOR with pattern = X'00' R1 = X'0000'	R1(1)	
0006	10- Same as 4		
0007	11- Same as 2 with pattern = X'00'		
	12- XOR with pattern = X'FF and NOR same pattern with R1 = X'00FF'		
0008	13- Same as 4		
0009	14- Same as 2 with pattern = X'00'	R1(0)	

DX1D - TRM INSTRUCTION TEST

This routine checks the appropriate condition latch after executing a Test Register Under Mask instruction.

ERC	Function	REG	RAC
	1- Load first operand = X'09' and execute TRM instruction with mask = X'05'	R1(1)	
0001	2- Test C latch		87F
0001	3- XOR with pattern = X'09' and test Z latch to verify if TRM instruction does not alter the register initial value		
	4- Set R1 = X'00FF' and execute TRM instruction with mask = X'FF'		
0002	5- Test CZ latches		
0003	6- Same as 3 with pattern = X'FF'		
	7- Test X'00' with X'FF'		
0004	8- Same as 5		
0005	9- Same as 3 with pattern = X'FF'	R1(0)	
	10- Set R1 = X'FF00' and execute TRM instruction with mask = X'FF'		
0006	11- Same as 5		
0007	12- Same as 3 with pattern = X'00'		

DX1E - SRI INSTRUCTION TEST

This routine checks for correct instruction decoding, correct action on the CZ latches and that the instruction does not alter the second operand.

ERC	Function	REG	RAC
	1- Subtract X'05' from X'09'	R1(1)	
0001	2- XOR with pattern = X'04' and test Z latch to verify result		87F
0002	3- Subtract X'00' from X'FF' and test CZ latches	1(0)	
0003	4- Same as 2 with pattern = X'FF'		
0004	5- Set CZ = 10, subtract X'00' from X'00' and test CZ latches		
0005	6- Same as 2 with pattern = X'00'		
0006	7- Set CZ = 10, subtract X'FF' from X'FF' and test CZ latches	R1(1)	
0007	8- Same as 2 with pattern = X'00'		
0008	9- Subtract X'FF' from X'00' and test CZ latches		
0009	10- Verify result by XORing byte 0 and byte 1 and testing CZ latches		

DX1F - CRI INSTRUCTION TEST

This routine checks that the compare does not alter the initial value in the register, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Compare X'05' with X'09' and test C latch	R1(1)	
0001	2- XOR with pattern = X'09' and test Z latch		87F
	3- Set R1 = X'00FF' and compare X'FF' with X'FF'		
0002	4- Test CZ latches		
0003	5- Same as 2 with pattern = X'FF'		
	6- Compare X'FF' with X'00'		
0004	7- Same as 4		
0005	8- Same as 2 with pattern = X'00'		
	9- Compare X'FE' with X'FF'	R1(0)	
0006	10- Same as 4		
0007	11- Same as 2 with pattern = X'FF'		

DX20 - LCR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09' and operand 2 = X'05', and execute LCR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding of operand 1 by XRI and test Z latch	R1(1)	87F
0002	3- Verify, using XRI, that operand 2 has not been altered and test the Z latch.	R3(1)	
0003	4- Set CZ = 10, R3 = X'xx01', move R3 low to R3 high, and test CZ latches	R3(1) R3(0)	
0004	5- Same as 2		
0005	6- Same as 4, with R3 = X'0002'		
0006	7- Same as 2		
0007	8- Same as 4, with R3 = X'0004'		
0008	9- Same as 2		
0009	10- Same as 4, with R3 = X'0008'		
000A	11- Same as 2		
000B	12- Same as 4, with R3 = X'0010'		
000C	13- Same as 2		
000D	14- Same as 4, with R3 = X'0020'		
000E	15- Same as 2		
000F	16- Same as 4, with R3 = X'0040'		
0010	17- Same as 2		
0011	18- Same as 4, with R3 = X'0080'		
0012	19- Same as 2		
0013	20- Same as 4, with R3 = X'007F'		
0014	21- Same as 2		
0015	22- Same as 4, with R3 = X'00BF'		
0016	23- Same as 2		
0017	24- Same as 4, with R3 = X'00DF'		
0018	25- Same as 2		
0019	26- Same as 4, with R3 = X'00EF'		
001A	27- Same as 2		
001B	28- Same as 4, with R3 = X'00F7'		
001C	29- Same as 2		
001D	30- Same as 4, with R3 = X'00FB'		
001E	31- Same as 2		
001F	32- Same as 4, with R3 = X'00FD'		
0020	33- Same as 2		
0021	34- Same as 4, with R3 = X'00FE'		
0022	35- Same as 2		
0023	36- Set CZ = 00, R3 = X'FE00', move R3 low to R3 high, and test CZ latches	R3(1) R3(0)	
0024	37- Same as 2		
0025	38- Same as 35, with CZ = 01, R3 = X'00FF'		
0026	39- Same as 2		
0027	40- Same as 35, with CZ = 01, R3 = X'00AA'		
0028	41- Same as 2		
0029	42- Same as 35, with CZ = 01, R3 = X'0055'		
002A	43- Same as 2		
	44- Set R1 = X'FF00', move R1 low to R3 low	R1(1)	
002B	45- Same as 2	R3(1)	
	46- Set R1 (1) = X'FF' and move R3 high to R1 high	R1(1)	
002C	47- Same as 2	R1(0) R3(0)	
	48- Move R3 high to R1 low		
002D	49- Same as 2		

DX22 - B, BCL, BZL AND BB INSTRUCTIONS TEST

This routine makes a positive and negative branch test.

ERC	Function	REG	RAC
	1- Set CZ = 11, R1 = X'0010'		
0001	2- Branch forward display for BZL		87F
0002	3- B Instruction negative display.		
0003	4- Negative display for BCL		
0004	5- Negative display for BZL		
0005	6- Alternate branch on bit and branch on error with positive and negative displacement		

DX23 - ACR INSTRUCTION TEST

This routine checks for correct instruction decoding, correct action on the CZ latches and that the instruction does not alter the second operand.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', and operand 2 = X'05', and execute ACR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'0E') and test the Z latch	R1(0)	87F
	3- Set R1 = X'F700' and R3 = X'7F01'		
	4- Add R3 low to R1 high	R1(0) R3(1)	
0002	5- Test CZ latches		
0003	6- XOR R1(0) with pattern = X'F8' then test CZ latches		
	7- Set R1 = X'FF81'		
0004	8- Add R3 high to R3 low, add R1 low to R3 high, then test C latch		
0005	9- Test Z latch		
0006	10- Same as 2, with pattern = X'80'	R3(1)	
0007	11- Same as 2, with pattern = X'00'	R3(0)	
0008	12- Same as 2, with pattern = X'81'	R1(1)	
0009	13- Same as 2, with pattern = X'FF'	R1(0)	
	14- Set R1 = X'FF00' and add R1 high to R1 high	R1(1)	
000A	15- Same as 5		
000B	16- Same as 2, with pattern = X'FE'	R1(0)	
	17- Set R3 = X'00FF', add R1 low to R3 low	R1(1)	
000C	18- Same as 2, with pattern = X'FF'	R3(1)	

DX24 - OCR INSTRUCTION TEST

This routine checks for correct action on the first operand and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute OCR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'0D') and test Z latch		87F
	3- Set R3 = X'000C', R1 = X'3300', AND OCR R3 low with R1 high	R3(1) R1(0)	
0002	4- Test CZ latches		
0003	5- Compare result in R1 (1) with X'FF' and test Z latch		
	6- Set R1 = X'FF00' and execute OCR R1 low with R3 high		
0004	7- Same as 4		
0005	8- Same as 5, with pattern = X'00'	R1(1)	
	9- Execute OCR R1 high with R1 high		
0006	10- Same as 5, with R1 (1)	R1(1)	

DX25 - NCR INSTRUCTION TEST

This routine checks for correct action on the first operand and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute NCR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'01') and test latch		87F
	3- Set R1 = X'00FF', R3 = X'FF00', and AND R1 low with R1 high	R1(0) R1(1)	
0002	4- Test CZ latches		
0003	5- Compare result in R1 (1) with X'FF' and test Z latch	R1(1)	
	6- Set CZ = 10 and AND R1 low with R1 high	R3(0)	
0004	7- Same as 4		
0005	8- Same as 5, with pattern = X'00'	R1(0)	

DX26 - XCR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute XCR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'0C') and test Z latch		87F
	3- Set R1 = X'FF00', R3 = X'0000', and OR R1 high with R1 low	R1(1) R1(0)	
0002	4- Test CZ latches		
0003	5- Compare result in R1 (1) with X'FF' and test Z latch		
	6- Set R3 (0) = X'FF' and OR R3 (0) with R1 (1)	R3(0)	
0004	7- Same as 4		
0005	8- Same as 5, with pattern = X'00'	R3(0)	

DX27 - SCR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and subtract	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'04') and test Z latch	R1(1)	87F
	3- Set R3 = X'00FF', R1 = X'FF00', and subtract R1 high from R3 high	R1(0) R3(0) R3(0)	
0002	4- Test CZ latches		
0003	5- Same as 2, with pattern = X'01'	R3(0)	
	6- Set R3 = X'FFFF' and subtract R1 high from R3 high	R1(0)	
0004	7- Same as 4		
0005	8- Same as 2, with pattern = X'00'	R3(0)	

DX28 - CCR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute CCR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'09') and test Z latch	R1(1)	87F
	3- Set R1 = X'FF00', R3 = X'00FF', and compare R3 low with R3 high	R3(0) R3(1)	
0002	4- Test CZ latches		
0003	5- Same as 2, with pattern = X'00'	R3(0)	
	6- Set R3 = X'00FF' and compare R3 high with R3 low		
0004	7- Same as 4		
0005	8- Same as 2, with pattern = X'FF'	R3(1)	
	9- Set R1 = X'01xx', R3 = X'02xx', and CZ = 01, and compare R3 high	R1(0) R3(0)	
0006	10- Test C latch		
0007	11- Test Z latch		

DX29 - LCOR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute LCOR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'02') and test Z latch		87F
	3- Set R3 = X'FF00', R1 = X'FF00', and load R3 high into R3 high		
0002	4- Test CZ latches		
0003	5- Same as 2, with pattern = X'7F'		
	6- Set R1 (0) = X'FF', CZ = 10, and load R1 low into R1 low		
0004	7- Same as 4		
0005	8- Compare R1 (1) with pattern = X'00' and test Z latch		

DX2A - LHR INSTRUCTION

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute LHR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'05') and test Z latch		87F

DX2B - SHR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and subtract R3 from R1	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'04') and test Z latch		87F
	3- Set R1 = X'0100', R3 = X'0000', and subtract R1 from R3	R1 R3	
0002	4- Test CZ latches		
0003	5- Same as 2, with pattern = X'FF'	R3(0)	
0004	6- Same as 2, with pattern = X'00'	R3(1)	
	7- Set R3 = X'FF00', R1 = X'FF00', and subtract R3 from R1	R1	
0005	4- Same as 4		
	9- Set R1 = X'01FF' and subtract R1 from R3	R1	
0006	10- same as 4		
0007	11- same as 2, with pattern = X'FD'	R3(0)	
0008	12- same as 2, with pattern = X'01'		

DX2C - CHR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and compare R3 with R1	R1 R3	
0001	2- Verify correct decoding by XRI (pattern = X'09') and test Z latch		87F

DX2E - DATA FLOW PATH BYTE 0 AND 1 USING LHR AND CHR (PART 1)

This routine tests the data flow path using the LHR and CHR instructions.

ERC	Function	REG	RAC
	1- Initialize R3 = X'FF00'	R3	
	2- Save current test pattern, set CZ = 01, and move R3 into R1	R3 R1	
0001	3- Test CZ latches		87F
0002	4- Verify correct transfer by XCR and test Z latch	R3(0)	
0003	5- Same as 4		
	6- Restore R3 and compare	R3	
0004	7- Same as 3		
	8- Same as 1		
	9- Update test pattern by adding 1 to R3(1) and subtracting 1 from R3(0)	R3(1) R3(0)	
	10- Test for end of test (255 passes)		

DX2F - DATA FLOW PATH BYTE 0 AND 1 USING LHR AND CHR (PART 2)

This routine tests the data flow path using the LHR and CHR instructions.

ERC	Function	REG	RAC
	1- Set R3 = X'FFFF', R1 = X'0000', CZ = 10, and R5(1) = X'FF'	R3,R1 R5(1)	
	2- Move R1 to R3		
0001	3- Test CZ latches		87F
	4- Set CZ = 10, R5(0) = X'FF', and compare R3 and R1	R5(0)	
0002	5- Same as 3		
	6- Compare R1 with R5 (R1 unchanged)	R1,R5	
0003	7- Same as 3		
0004	8- Verify correct transfer using XRI (pattern = X'00') and test the Z latch.	R3(1)	
0005	9- Same as 8	R3(0)	

DX31 - AHR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute AHR instruction	R3(1) R1(1)	
0001	2- Verify correct decoding by XRI (pattern = X'0E') and test Z latch	R1(1)	87F
	3- Set R1 = X'0000', CZ = 10, and add R1 to R1		
0002	4- Test CZ latches		
0003	5- Set R5 = X'0100', R3 = X'FF00', add R5 to R3, and test Z latch	R5,R3	
0004	6- Test C latch		
0005	7- compare R1 with R3 and test Z latch	R1,R3	
	8- Set R1 = X'FFE1', R3 = X'0001', CZ = 10, and add R3 to R1		
0006	9- Same as 4		
0007	10- Same as 2, with pattern = X'FF'	R1(1)	
0008	11- Same as 2, with pattern = X'FF'	R1(0)	

DX32 - OHR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute OHR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'0D') and test Z latch	R1(1)	87F
	3- Set R5 = X'55AA', R1 = X'55AA', CZ = 01, and OR R5 with R5	R5,R5	
0002	4- Test CZ latches		
0003	5- Compare R5 with R1 and test Z latch		
	6- Set R3 = X'AA55', R1 = X'00AA', CZ = 01, and OR R5 with R3	R5,R3	
0004	7- Same as 4		
0005	8- Same as 2, with pattern = X'FF'	R5(0)	
0006	9- Same as 2, with pattern = X'FF'	R5(1)	
	10- Set R1 = X'0000', CZ = 10, and OR R1 with R1		
0007	11- Same as 4		
0008	12- Verify correct decoding by CRI (pattern = X'00') and test Z latch	R1(0)	
0009	13- Same as 12	R1(1)	

DX33 - NHR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute NHR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'01') and test Z latch	R1(1)	87F
	3- Set R1 = X'AA55', R5 = X'FFFF', CZ = 01, and AND R1 with R5	R1,R5	
0002	4- Test CZ latches		
0003	5- Verify correct decoding by CRI (pattern = X'AA') and test Z latch	R1(0)	
0004	6- Same as 5, with pattern = X'55'	R1(1)	
	7- Set R5 = X'55AA', (XOR R1 with R5), and OR R1 with R5	R1,R5	
0005	8- Same as 4		
0006	9- Same as 5, with pattern = X'00'	R1(0)	
0007	10- Same as 9	R1(1)	
	11- Set R3 = X'FFFF' and AND with R5	R3,R5	
0008	12- Same as 4		
0009	13- Same as 5, with pattern = X'55'	R3(0)	
000A	14- Same as 5, with pattern = X'AA'	R3(1)	

DX34 - XHR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute XHR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'0C') and test Z latch		87F
	3- Set R5 = X'FFFF', R3 = X'AA55', R1 = X'0000', CZ = 01, and XOR R1 with R3	R1,R5	
0002	4- Test CZ latches		
0003	5- Verify correct decoding by CRI (pattern = X'AA') and test Z latch	R1(0)	
0004	6- Same as 5, with pattern = X'55'	R1(1)	
	7- XOR R1 with R5	R1,R5	
0005	8- Same as 4		
0006	9- Same as 5, with pattern = X'55'		
0007	10- Same as 5		
	11- Set CZ = 10, R1 = X'55AA', and XOR R1 with R1		
0008	12- Same as 4		
0009	13- Same as 5, with pattern = X'00'	R1(0)	
000A	14- Same as 13	R1(1)	
	15- Set CZ = 10, R3 = X'AA55', and XOR R3 with R3	R3	
000B	16- Same as 4		
000C	17- Compare R3 with R1 and test Z latch		

DX35 - LHOR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', set R3 = X'0005' and execute LHOR instruction R3 into R1	R1(1) R3	
0001	2- Verify correct decoding by XRI (pattern = X'02') and test Z latch		87F
	3- Set R1 = R3 = X'0102', load, and shift (LHOR) R1 into R1	R1	
0002	4- Test CZ latches		
0003	5- Verify correct decoding by CRI (pattern = X'00') and test Z latch	R1(0)	
0004	6- Same as 5, with pattern = X'81'		
	7- LHOR R1 into R1	R1	
0005	8- Same as 4		
0006	9- Same as 5	R1(0)	
0007	10- Same as 5, with pattern = X'40'	R1(1)	
	11- Set R1 = X'0001' and CZ = 00 by LHOR instruction, execute LHOR instruction R1 into R1	R3,R1	
0008	12- Test C latch		
0009	13- test Z latch		
000A	14- Verify result in R1 by OHR instruction and test Z latch		

DX36 - LOR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute LOR instruction R3 into R1	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'02') and test Z latch	R1(1)	87F

DX37 - AR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and add R3 to R1	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'0E') and test Z latch	R1(1)	87F

DX38 - DATA FLOW PATH BYTE X PATTERN SENSITIVITY TEST

This routine uses the LOR and AR instructions to test the byte X data flow path.

ERC	Function	REG	RAC
	1- Clear R1 byte X, set R1 = X'00001', CZ = 10, and execute LOR instruction	R1	
0001	2- Test C latch		87F
0002	3- Test Z latch		
0003	4- Verify result in R1 (OHR instruction) and test Z latch	R1	
	5- Set R1 = X'C400', CZ = 01, and add R1 to R1; expected result R1 = X'018800'	R1	
0004	6- Test CZ latches		
0005	7- Verify correct decoding by CRI (pattern = X'88') and test Z latch	R1(0)	
0006	8- Same as 7 with pattern = X'00'	R1(1)	
	9- Set R3 = X'0000', CZ = 01, shift, and load R1 into R3; expected result R3 = X'xxC400'		
0007	10- Same as 6		
0008	11- Same as 7 with pattern = X'C4'	R3(0)	
0009	12- Same as 7 with pattern = X'00'	R3(1)	
	13- Set R1 = X'310000' (by five successive adds of R1 to R1) CZ = 00	R1,R3	
000A	14- Same as 6		
000B	15- Same as 4	R1	
	16- Set CZ = 01, R3 high = X'00', do 5 shifts, load R1 into R3, and successively shift R3 into R3; expected result R3 = X'00C400'	R1,R3	
000C	17- Same as 6		
	18- Set R3 = X'AAA0' and R3 = X'2AA800' (by six adds R3 to R3)		
000D	19- Same as 6		
	20- Set R1 = X'557F', CZ = 10, shift, and load R3 into R1; expected result R1 = X'155400'	R1,R3	
000E	21- Same as 6		
000F	22- Same as 7 with pattern = X'54'	R1(0)	
	23- Shift and load R1 into R1		
0010	24- Same as 7 with pattern = X'AA'		
	25- Shift and load R1 into R1 six times, CZ = 00; expected result R1 = X'02AA80'	R1	
0011	26- Same as 7 with pattern = X'2A'	R1(0)	
0012	27- Same as 7 with pattern = X'A8'		

DX3A - LA INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X.

ERC	Function	REG	RAC
	1- Clear R1 and load address X'0509'	R1	
0001	2- Verify correct decoding by XRI (pattern = X'05') and test Z latch	R1(0)	87F
	3- Set CZ = 10 and load address X'000000'	R3	
0002	4- Test CZ latches		
0003	5- Set R1 = X'0000', compare R3 with R1, and test Z latch	R3,R1	
0A00	6- Go to subroutine 'SBXT' to test byte X; expected result = X'00'	R0,R1 R3,R5	
	7- Set CZ = 01, load address X'1FFFFF'		
0004	8- Same as 4		
0005	9- Set R3 = X'FFFF', compare R3 with R1, and test Z latch	R1,R3	
	10- Reset R1 high; R1= X'1F00FF'	R1(0)	
0A00	11- Same as 6 expected result = X'F8'	R1	
	12- Load address X'1555AA' into R1 and set R3 = X'55AA'	R1,R3	
0006	13- Compare R1 with R3 and test Z latch	R1,R3	
0A00	14- Same as 6, with expected result = X'AA'		
	15- Same as 12, address = X'1AAA55', R3 = X'AA55'	R1,R3	
0007	16- Same as 13		
0A00	17- Same as 6, with expected result = X'D5'		

DX3B - DATA FLOW PATH BYTE X, 0 AND 1

This routine uses the LA instruction. It loops on a data table to load successive halfwords to verify the data flow path for bytes 0 and 1. Byte X is tested by calling subroutine 'SBXT'

ERC	Function	REG	RAC
	This routine loops forty times with LA instruction being updated on each pass. Use register 7 as base register	R7	
0001	1- The data in R1 (loaded by LA instruction) is tested by compare R1 with R3 (R3 is loaded via test table) and test Z latch	R1,R3	87F
0A00	2- Go to subroutine 'SBXT' to test byte X; expected result is loaded from first byte of each word of the data table (STBL)		

DX3C - LR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute LR to move R3 into R1	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'05') and test Z latch	R1(1)	87F
	3- Set R7 = X'000000', R1 = X'3FFFFF', CZ = 10, and load register R7 into R1	R7,R1	
0002	4- Test CZ latches		
0003	5- Verify byte 0 and 1 of R1 by comparison with R3 = X'0000' and test Z latch	R1,R3	
0A00	6- Go to subroutine SBXT to test byte X; expected result = X'00'	R1(X)	
	7- Set R7 = X'2AAA55', CZ = 01, and load register R7 into R1	R1,R7	
0004	8- Same as 4		
0005	9- Same as 5, with R3 = X'AA55'		
0A00	10- Same as 6, with expected result = X'AA'	R1(X)	
	11- Same as 7 with R7 = X'1555AA'		
0006	12- Same as 4		
0007	13- Same as 5, with R3 = X'55AA'	R1,R3	
0A00	14- Same as 6, with expected result = X'55'	R1(X)	

DX3D - LOCAL STORE REGISTER 3 AND 5 BYTE X TEST

This routine checks the correct loading of byte X by shifting it into byte 0 using macro RBXCL and testing. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Set R7 = X'3F0000'	R1(1) R3(1)	
0001	2- Move R7 into R3, shift byte X into byte 0, compare R3(0) with pattern = X'FC', and test Z latch		87F
0002	3- Same as 2, with R5	R1,R3	
	4- Same as 1, with pattern = X'000000'	R1,R5	
0003	5- Same as 2, compare with pattern = X'00'	R1(X)	
0004	6- Same as 5, with R5		
	7- Same as 1, with pattern = X'2A0000'		
0005	8- Same as 2, compare with pattern = X'A8'		
0006	9- Same as 8, with R5		
	10- Same as 1, with pattern = X'1B0000'		
0007	11- Same as 2, compare with pattern = X'54'		
0008	12- Same as 11, with R5		

DX3E - OR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute OR instruction	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI instruction (pattern = X'0D') and test Z latch	R1(1)	87F
0002	3- Set R1, R3, R5 = X'000000' (via LA instruction), CZ = 10, OR with R1, and test CZ latches	R1,R3	
0003	4- Compare R1 with R5 (byte 0 and 1) and test Z latch	R1,R5	
0A00	5- Go to subroutine 'SBXT' to test byte X; expected result = X'00'	R1(X)	
0004	6- Same as 3, with R1 = X'2555AA', R3 = X'1AA55', R5 = X'3FFFFF', and CZ = 01		
0005	7- Same as 4		
0A00	8- Same as 5, expected result = X'FF'		
0006	9- Same as 3, with R1 = X'1F8FFF', R3 = X'2FFFFF', R5 = X'3FFFFF', and CZ = 01		
0007	10- Same as 4		
0A00	11- Same as 5, expected result = X'FF'		
0009	12- Same as 3, with R1, R3, and R5 = X'300000', CZ = 01		
000A	13- Same as 4		
0A00	14- Same as 5, expected result = X'FC'		
000B	15- Same as 3, with R1 = X'00AA55', R3 = X'0055AA', R5 = X'00FFFF', and CZ = 01		
000C	16- Same as 4		
0A00	17- Same as 5, expected result = X'03'		

DX3F - NR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and execute NR instruction		
0001	2- Verify correct decoding by XRI (pattern = X'01') and test Z latch		87F
0003	3- Set R1, R3, and R5 = X'3F0000', CZ = 01, AND R1 with R3, and test CZ latches	R1,R3	
0004	4- Compare R1 with R5 (byte 0 and 1) and test Z latch	R1,R5	
0A00	5- Go to subroutine 'SBXT' to test byte X; expected result = X'FC'	R1(X)	
0005	6- Same as 3, with R1 = X'2AAA55', R3 = X'1555AA', R5 = X'00000', and CZ = 10		
0006	7- Same as 4		
0A00	8- Same as 5, expected result = X'00'		
0007	9- Same as 3, with R1 = X'1555AA', R3 = X'2AAA55', R5 = X'000000', and CZ = 10		
0008	10- Same as 4		
0A00	11- Same as 5, expected result = X'00'		
0009	12- Same as 3, with R1, R3, and R5 = X'00FFFF', CZ = 01		
000A	13- Same as 4		
0A00	14- Same as 5, expected result = X'03'		

DX40 - XR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and XOR operand 2 with operand 1	R1,R3	
0001	2- Verify correct decoding by XRI (pattern = X'0C'), test Z latch		87F
0002	3- Set R1 and R3 = X'3FAA55', R5 = X'000000' (via LA instruction) set CZ = 10, XOR R1 with R3, and test CZ latches	R1,R3	
0003	4- Compare HW R1 with R5 and test Z latch	R1,R5	
0A00	5- Go to subroutine X'SBXT' to test byte X; expected result = X'00'	R1(X)	
0005	6- Same as 3, with R1 = X'1555AA', R3 = X'1555AA', R5 = X'3F0000', and CZ = 01		
0006	7- Same as 4		
0A00	8- Same as 5, expected result = X'FC'		
0008	9- Same as 3, with R1 = X'15AA55', R3 = X'2AAA55', R5 = X'3F0000', and CZ = 01		
0009	10- Same as 4		
0A00	11- Same as 5, expected result = X'F9'		
000A	12- Same as 3, with R1 = X'00AA55', R3 = X'0055AA', R5 = X'00FFFF', and CZ = 01		
000B	13- Same as 4		
0A00	14- Same as 5, expected result = X'03'		

DX41 - AR INSTRUCTION TEST (OVERFLOW TEST)

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Set R1 = X'2AAA55', R3 = X'3555AA', R5 = X'1FFFFF', and CZ = 01	R1,R3	
0001	2- Test CZ latches		87F
0003	3- Verify correct decoding by XHR and test Z latch	R1,R5	
0A00	4- Go to subroutine X'SBXT' to test byte X, expected result = X'7C'	R1(X)	
	5- Same as 1, with R1 = X'1555AA', R3 = X'2AAA56', R5 = X'000000', and CZ = 10		
0004	6- Test C latch		
0005	7- Test Z latch		
0006	8- Same as 3		
0A00	9- Same as 4, expected result = X'00'		
	10- Same as 1, with R1 = X'1FFFFF', R5 = X'3FFFFE', CZ = 01		
0008	11- Same as 2		
0009	12- Same as 3		
0A00	13- Same as 4, expected result = X'FC'		

DX42 - SR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', subtract R3 from R1	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'04') and test Z latch		87F
0002	3- Set R1 = X'1555AA', R3 = X'2AAA55', R5 = X'2AAB55', CZ = 01, subtract R3 from R1, and test CZ latches	R1,R3	
0003	4- Verify correct decoding by XHR and test Z latch	R1,R5	
0A00	5- Go to subroutine 'SBXT' to test byte X, expected result = X'A8'	R1(X)	
0004	6- Same as 3, with R1, R3 = X'3FFFFF', R5 = X'000000', CZ = 10		
0005	7- Same as 4		
0A00	8- Same as 5, expected result = X'00'		
0006	9- Same as 3, with R1 = X'0055AA', R3 = X'1AAA55', R5 = X'25AB55', and CZ = 01		
0007	10- Same as 4		
0A00	11- Same as 5, expected result = X'96'		
0008	12- Same as 3, with R1 = X'00AA55', R3 = X'0055AA', R5 = X'0054AB', and CZ = 01		
0009	13- Same as 4		
0A00	14- Same as 5, expected result = X'00'		

DX43 - CR INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Load operand 1 = X'09', operand 2 = X'05', and compare second with the first	R1(1) R3(1)	
0001	2- Verify correct decoding by XRI (pattern = X'09') and test Z latch	R1(1)	87F
0002	3- Load R1 = R5 = X'155555', R3 = X'2AAAAA', CZ = 01, compare R3 with R1, and test CZ latches	R1,R3 R5	
0003	4- Verify correct decoding by XHR and test Z latch	R1,R5	
0A00	5- Go to subroutine 'SBXT' to test byte X; expected result = X'A4'	R1(X)	
0004	6- Same as 3, with R1 = R3 = R5 = X'3F55AA', CZ = 10		
0005	7- Same as 4		
0A00	8- Same as 5, expected result = X'FC'		
0007	9- Same as 3, with R1 = R5 = X'255555', R3 = X'1AAAAA', CZ = 10		
0008	10- Same as 4		
0A00	11- Same as 5, expected result = X'94'		
000A	12- Same as 3, with R1 = R5 = X'00AAAA', R3 = X'01AAAA', CZ = 01		
000B	13- Same as 4		
0A00	14- Same as 5, expected result = X'00'		
000C	15- Same as 3, with R1 = R5 = R3 = X'00AA55', CZ = 10		
000D	16- Same as 4		
0A00	17- Same as 5, expected result = X'00'		

DX44 - L INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	R7 is used as base register	R1,R3	
0001	1- Load, via LA instruction, R1 = X'255AA5' (background data), R3 = X'1AA55A', CZ = 01, load pattern = X'1AA55A' into R1, and test CZ latches		87F
0002	2- Compare R1 to R3 and test Z latch	R1,R3	
0003	3- Same as 1, with R1 = X'1AA55A', R3 = X'1A5AA5', CZ = 01, pattern = X'255AA5'		
0004	4- Same as 2		
0005	5- Same as 1, with R1 = X'3FFFFF', R3 = X'000000', CZ = 10, pattern = X'000000'		
0007	6- Same as 2		

DX45 - LH INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	R7 is used as base register		
0001	1- Load R1 = X'2AA55A' (background data), R3 = X'005AA5', CZ = 01, load HW pattern = X'5AA5' into R1, and test CZ latches	R1,R3	87F
0002	2- Compare R1 to R3 and test Z latch		
0003	3- Same as 1, with R1 = X'155AA5', R3 = X'00A55A', CZ = 01, pattern = X'A55A'		
0004	4- Same as 2		
0005	5- Same as 1, with R1 = X'3FFFFF', R3 = X'000000', CZ = 10, pattern = X'0000'		
0006	6- Same as 2		

DX46 - STH INSTRUCTION TEST

This routine checks for correct action on the CZ latches, and for correct stored data. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	R7 is used as base register		
0001	1- Load background data in test area, execute STH instruction, set R1 = X'00A55A', CZ = 01, store, and test CZ latches	R1,R3	87F
0002	2- Load data stored above (via L instruction) compared and test Z latch	R1,R3	
0003	3- Same as 1, with R1 = X'005AA5', CZ = 10		
0004	4- Same as 2		

DX47 - L AND LH TEST USING R0 AS A SINK

This routine checks for correct action on the CZ latches, using R0 as the operand.

ERC	Function	REG	RAC
	R3 is used as base register		
0001	1- Set CZ = 01 and load instruction with R0 as first operand	R0	87F
0002	2- Test CZ latches		

DX48 - L TEST (FROM FW DIRECT ADD. SAVE AREA)

This routine checks for correct action on the CZ latches, and for correct moved data. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Load background data into R1, R3 = X'3FFFFF', CZ = 01, load R1 from direct addressable area, R1 = X'3FFFFF'	R1,R3	
0001	2- Test CZ latches		87F
0002	3- Compare R1 with expected data and test Z latch	R1,R3	

DX49 - LR USING R0 AS THE SINK

This routine checks for a correct branch, and for correct action on the CZ latches.

ERC	Function	REG	RAC
0001	1- Load R5 with correct branch address, set CZ = 01, and execute LR instruction with R0 as first operand	R0,R5	87F
0002	2- Test CZ latches		

DX4A - IC INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	R3 is used as base register	R1,R7	
	Loop to save HW Direct Addressable Save Area into data table		
	Loop to store load data table into HW Direct Addressable Area		
	1- Load background data into R1, expected result into R7 = X'3F0055', CZ = 01, and execute IC instruction		
0001	2- Test CZ latches		87F
0002	3- Compare result and test Z latch	R1,R7	
	4- Same as 1, with R7 = X'3F00FF', CZ = 10		
0003	5- Test Z latch		
0004	6- Test C latch		
0005	7- Same as 3		
	8- Same as 1, with R7 = X'3F01FF', CZ = 10		
0006	9- Same as 2		
0007	10- Same as 3		
	11- Same as 1, with R7 = X'0000AA', CZ = 01		
0008	12- Same as 2		
0009	13- Same as 3		

DX4B - ICT INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	R3 is the base register A data table is used R3 is loaded with the address of the data table + 122		
	1- Load background data into R1 and expected result into R7 = X'2A0055', CZ = 01	R1,R7	
0001	2- Execute ICT instruction and test CZ latches	R1(1) R3	87F
0002	3- Compare result in R1 and test Z latch	R1,R7	
0003	4- Load R7 with address of data table + 123, compare result in R3, and test Z latch	R3,R7	
	5- Same as 1, with R7 = X'15AAFF', CZ = 10		
0004	6- Same as 2		
0005	7- Same as 3		
0006	8- Same as 4, with address of data table + 124		
	9- Same as 1, with R7 = X'00FF00', CZ = 10		
0007	10- Same as 2		
0008	11- Same as 3		
0009	12- Same as 4, with address of data table + 125		
	13- Same as 1, with R7 = X'3FFF00', CZ = 01		
000A	14- Same as 2		
000B	15- Same as 3		
000C	16- Same as 4, with address of data table + 126		

DX4C - ST INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

Test for zeros stored in the highest order bits of byte X.

ERC	Function	REG	RAC
	R3 is used as base register		
0001	1- Load R1 = X'15A55A', CZ = 01, store R1 in area test, and test CZ latches	R1	87F
0002	2- Load R7 = stored data, compare R1 with R7, and test Z latch	R1,R7	
0003	3- Same as 1, with R1 = X'2A5AA5' CZ = 10		
0004	4- Same as 2		
0005	5- Same as 1, with R1 = X'3FFFFF', and CZ = 01 (fullword direct addressable save area)		
0006	6- Same as 2		
0007	7- Same as 5, R1 = X'000000', CZ = 10		
0008	8- Same as 2		
	9- Load R1 = X'3FFFFF' into test area		
	10- Load expected data R7 = X'FFC0'		
0009	11- Load first HW of FW stored, compare R5 with R7, and test Z latch		

DX4D - STH TEST (USING HW DIRECT ADD. SAVE AREA)

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	R7 is used as base register Load background data is test area		
0001	1- Load R1 = X'00FFFF', CZ = 01, execute STH instruction to store R1, and test CZ latches	R1	87F
0002	2- Load R3 = stored data, R1 = X'3FFFFF', compare registers, and test Z latch	R1,R3	
0003	3- Same as 1, with R1 = X'000000', CZ = 10		
0004	4- Same as 2, with R1 = X'3F0000'		

DX4E - STC INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

A test area and the byte direct addressable area are used.

ERC	Function	REG	RAC
	R3 is used as base register Load background data in test area and Byte Direct Addressable Area		
0001	1- Load stored data in R1 = X'3FAAFF', set CZ = 01, store character in test area, and test CZ latches	R1 R1(1)	87F
0002	2- Load stored data, compare with expected data = X'00FF55', and test Z latch	R1,R7	
0003	3- Same as 1, with R1 = X'3FFF55', CZ = 10		
0004	4- Same as 2, with expected data = X'0055AA'		
0005	5- Same as 1, with R1 = X'2AAAFF', CZ = 10, with byte direct addressable area		
0006	6- Same as 2, with expected data = X'00FFFF'		
0007	7- Same as 5, R1 = X'3F00AA', CZ = 10		
0008	8- Same as 2, with expected data = X'0000FF'		

DX4F - STCT INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

A test area is used.

ERC	Function	REG	RAC
	R5 is used as base register Load background data into test area and register		
0001	1- Load test data R1 = X'3FFF55', set CZ = 01, load address table into base register, execute STCT instruction, and test CZ latches	R1(0) R3	87F
0002	2- Load R7 = stored data R1 = X'00FFAA' (expected data), and test Z latch	R1,R7	
0003	3- Compare R7 (= updated address) with R1, and test Z latch		
0004	4- Same as 1, with R1= X'3FFF00', CZ = 10		
0005	5- Same as 2, with R1 = X'00FF00'		
0006	6- Same as 3		

DX50 - SHIFT RIGHT FULLWORD TEST (PART 1 OF 2)

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
0001	1- Set R1 = X'2AAAAA', R3 = X'155555' (expected result), set CZ = 01, execute LOR instruction, and test CZ latches	R1,R3	87F
0002	2- Compare result in R1 with R3 and test Z latch	R1,R3	
0003	3- Same as 1, with R3 = X'0AAAAA', CZ = 10		
0004	4- Same as 2		
0005	5- Same as 1, with R3 = X'055555'		
0006	6- Same as 2		
0007	7- Same as 1, with R3 = X'02AAAA'		
0008	8- Same as 2		

DX51 - SHIFT RIGHT FULLWORD TEST (PART 2 OF 2)

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
0001	1- Set R1 = X'000001', load shift result R3 = X'000000', set CZ = X'00', execute LOR instruction, and test CZ latches	R1,R3	87F
0002	2- Compare result in R1 and test Z latch	R1,R3	

DX53 - 22 BITS ARI TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function	REG	RAC
0001	1- Set R1 = X'1FFFFF', R3 = X'200000' (expected result), add one to R1, compare R1 with R3, and test Z latch	R1,R3	87F

DX54 - 22 BITS SRI TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
0001	1- Reset R1, set R3 = X'FFFFF' (expected result), subtract one from R1, compare R1 with R3, and test Z latch	R1,R3 R1(1)	87F

DX55 - 22 BITS ACR TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
0001	1- Set R1 = X'2FFEFF', R3 = X'000300', R5 = X'3001FF' (expected result), add character register, compare, and test Z latch	R1,R3 R1(0) R3(0)	87F

DX56 - 22 BITS SCR TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
0001	1- Set R1 = X'3F0000', R5 = X'2FFFFF' (expected result), R3 = X'000001', subtract character register, compare, and test Z latch	R1,R5	87F

DX57 - BAL AND BALR INSTRUCTION TEST

This routine checks for a correct branch and for correct action on the CZ latches.

ERC	Function	REG	RAC
	1- Load R1 to R7 with branch stop address		
0001	2- Set CZ = 01 and execute BAL instruction to 'branch and test link continued'	R3	87F
0002	3- Test CZ latches		
0003	4- Get expected link in R1, compare R3 with R1, and test Z latch	R1,R3	87F
0004	5- Update return address, CZ = 10, and execute BALR instruction	R3(1)	
0005	6- Test CZ latches		
0006	7- Same as 3		
0007	8- Check branch address		

DX58 - BCT INSTRUCTION TEST

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Set R3 = X'3F0000' (BCT count), R7 = X'3FFFFF' (expected result), CZ = 01, and execute BCT instruction	R3,R7 R3(1)	
0001	2- Above BCT did not branch		87F
0002	3- Compare R1 with R7 and test Z latch (BCT did not decrement)	R1,R7	
0003	4- If above BCT branch test CZ latches		
0004	5- Same as 3 6- Set R7 = X'3FFEFF', CZ=10, execute BCT instruction		
0005	7- Same as 2		
0006	8- Same as 3		
0007	9- Same as 4		
0008	10- Same as 3	R1,R5	
	11- Set R1 = X'3F0001' (BCT count), R5 = X'3F0001' (expected result), CZ = 10, execute BCT instruction		
0009	12- Test CZ latches	R1(1)	
000A	13- Compare R5 with R1 and test Z latch (BCT did not decrement R5)		
000B	14- Above BCT did not branch	R3,R1	
	15- Set R3 = X'3F01FF' (BCT count), R1 = X'3F00FF' (expected result), CZ = 10		
000C	16- Test CZ latches		
000D	17- Compare R3 with R1 and test Z latch	R3,R1	
000E	18- BCT branched when byte 0 of R3 EQ 0		
000F	19- Compare R3 with R1 and test Z latch		

DX5A - REGISTER DECODE TEST, CURRENT INT LEV REGISTER GROUP (PART 1)

This routine checks for correct register decoding, and for correct action on the other registers. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Clear R1 through R7, set R1 = X'000001', load R1 into R1	R1	
0001	2- Check (OR instruction) all others registers and test Z latch		87F
0002	3- Load R1 into R1, compare, (pattern = X'01'), and test Z latch	R1(1)	
	4- Set R1 = X'3FFEFE', set low byte of R1 = X'01', and use high and low byte register decode	R1	
0003	5- Same as 2		
0004	6- Compare (pattern = X'01') and test Z latch	R1(1)	
0005	7- Same as 6	R1(0)	
	8- Clear R1, set R2 = X'000002', load R2 into R2	R1,R2	
0006	9- Same as 2		
0007	10- Move R2 into R1, compare (pattern = X'02'), and test Z latch		
	11- Clear R1 and R2, set R3 = X'000003', load R3 into R3	R1,R2	
0008	12- Same as 2		
0009	13- Same as 6, with pattern = X'03'	R3(1)	
	14- Set R3 = X'3FFCFC', set low byte of R3 = X'03', and load low byte into high byte of R3	R3	
000A	15- Same as 2		
000B	16- Same as 6, with pattern = X'03'	R3(1)	
000C	17- Same as 16	R3(0)	
	18- Clear R3, set R4 = X'000004' (via LA instruction), load R4 into R4	R3,R4	
000D	19- Same as 2		
000E	20- Move R4 into R1, compare (pattern = X'04'), and test Z latch	R1(1)	

DX5B - REGISTER DECODE TEST, CURRENT INTP LEV REGISTER GR (PART 2)

This routine checks for correct register decoding, and for correct action on the other registers. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
	1- Clear R1 through R7, set R5 = X'000005', load R5 into R5	R5	
0001	2- Check (via OR instruction) all others registers and test Z latch		87F
0002	3- Compare (pattern = X'05') and test Z latch	R5(1)	
	4- Set R5 = X'2AFAFA', set low byte of R5 = X'05', use high and low byte register decode	R5(1) R5(0)	
0003	5- Same as 2		
0004	6- Same as 3		
0005	7- Same as 3		
	8- Clear R5, set R6= X'000006', load R6 into R6	R5,R6	
0006	9- Same as 2		
0007	10- Move R6 into R1, compare (pattern = X'06'), and test Z latch	R1(1)	
	11- Clear R1 and R6, set R7 = X'000007', load R7 into R7	R1,R6	
0008	12- Same as 2	R7	
0009	13- Same as 3, with pattern = X'07'		
	14- Set R7 = X'00F8F8', set low byte of R7 = X'07', use high and low byte register decode	R7(1) R7(0)	
000A	15- Same as 2		
000B	16- Same as 3, with pattern = X'07'	R7(1)	
000C	17- Same as 3, with pattern = X'07'	R7(0)	

'5C ..ND SUBTRACT PATTERN SENSITIVITY TEST

This routine loops using the BCT instruction. It increments one register and decrements another, and then compares them with the value in the BCT instruction. The LA instruction is used to load bytes X, 0, and 1.

ERC	Function	REG	RAC
0001	1- Clear R1 and R3, set R7 = X'3FFFFE', R5 = X'3FFFFF', add one to R7(1), CZ = 00, branch and count with R3(1), and test CZ latches	R1,R3 R7	87F
0002	2- If effective branch test CZ latches		
0003	3- Execute XHR instruction R3 with R5, compare R1 with R3, and test Z latch	R3,R1 R5	
0004	4- XOR R7 with R5 and R7 with R1, and test Z latch		
	5- Restore (in complement form) SRI count and BCT count, update ARI count		
0005	6- Test CZ latches		
0006	7- Set CZ=10, decrement SRI count, and test C latch	R7(1)	
0007	8- OR R7 with R7 and test Z latch (Z latch set with non-zero SRI)	R7	
	9- End of loop		
0008	10- Increment R1 by one and test Z latch (test of overflow)	R1(1)	
0009	11- Test C latch (test of overflow)		
000A	12- XOR halfword R3 with R1 (verify that counts match)		
000B	13- Decrement R7 by one and test CZ latches		
000C	14- Complement R7 with R5, XOR R7 with R1 (add and subtract match), and test Z latch		

DX5F - INPUT/OUTPUT INSTRUCTION DECODE TEST

This routine tests the Out X'79' and In X'79' instructions, with the test running in levels 1, 2, 3, and 4.

Note: DX5F stands for DA5F, DB5F, DC5F, and DD5F. **Not DE5F.**

ERC	Function	REG	RAC
	1- Load R1 = X'000300', R7= X'3F0300', R3 with background data, CZ = 01 (expected level 5 CZ = 11)	R1,R3 R5,R7	
0001	2- Load R7 with Output X'79' and test CZ latches	R7	87F
0002	3- Load R3 with Input X'79' and test CZ latches	R3	
0003	4- Verify R3 (compare with R1) and test Z latch	R3,R1	
	5- Same as 1, with CZ = 10, R1 = R7 = X'000000' (expected level 5 CZ = X'00')		
0004	6- Same as 2		
0005	7- Same as 3		
0006	8- Same as 4		

DA60 - INPUT TEST FOR CCU LAG ADDR REGISTER

This routine checks for the correct loading of the LAR without a program check.

Other conditions are tested in the level change routine (level 2 to 1, level 5 to 1).

ERC	Function	REG	RAC
	1- Load expected address in R3	R3	
	2- Input (X'74') LAR in R1 and compare input address with expected address in R3	R1,R3	
0001	3- Test Z latch		87F

DA61 - GENERAL PURPOSE REGISTER INTERACTION TEST (LEVEL 1 ONLY)

The General Purpose registers for level 2, 3, 4 and 5 that were initialized by routine DA01 are tested to check that their contents were not altered by the DAxx routines running at level 1.

Note: This routine cannot be specifically selected. It checks that the general purpose register contents set by routines DA01, DB02, DC03, and DD04 have not changed after the CCU instruction set is tested.

ERC	Function	REG	RAC
	1- Go to subroutine 'SGRI'. Parameters: starting - Output = X'00', ending - Input = X'1F', expected data table		
0001	2- Check for interaction between level 1 and other general purpose registers.		886

DA62 - I/O REGISTER DECODE TEST (LEVEL 1 ONLY)

This routine uses subroutine 'SIOD' to test I/O register decoding, using the General Purpose registers (each General Purpose register, from level 1, register 6 through level 5, register 7, is tested).

ERC	Function	REG	RAC
	1- Go to subroutine X'SIOD'. Parameters: starting - Output = X'26', ending - Input = X'1F'		
0001	Either an Output or an Input register decode failure occurred		87F

63 ERAL PURPOSE REGISTER DATA SENSITIVITY (LEVEL 1 ONLY)

This routine uses subroutine 'SLST'. Each of the General Purpose registers tested above in routine DA62 is tested again, using 44 different patterns.

ERC	Function	REG	RAC
	1- Go to subroutine 'SLST'. Parameters: starting - Output = X'26', ending - Output = X'1F', table address, table length = X'2E'		
0001	2- Local store register failed		87F

DB67 - GENERAL PURPOSE REGISTER INTERACTION TEST (LEVEL 2 ONLY)

The General Purpose registers for level 1, 3, 4 and 5 that were initialized by routine DB02 are tested to check that their contents were not altered by the DBxx routines running at level 2.

Note: This routine cannot be specifically selected. It checks that the general purpose register contents set by routines DA01, DB02, DC03, and DD04 have not changed after the CCU instruction set is tested.

ERC	Function	REG	RAC
	1- Go to subroutine 'SGRI'. Parameters: starting - Output = X'08', ending - Input = X'27', expected data table		
0001	2- interaction between level 2 and some other general registers		886

DB69 - I/O REGISTER DECODE TEST (LEVEL 2 ONLY)

This routine uses subroutine 'SIOD' to test I/O register decoding, using the General Purpose registers (each General Purpose register, from level 2, register 6 through level 1, register 7, is tested).

ERC	Function	REG	RAC
	1- Go to subroutine 'SIOD'. Parameters: starting - Output = X'06', ending - Output = X'27'		
0001	2- Either an Output or an Input register decode failure occurred		87F

DB6A - GENERAL PURPOSE REGISTER DATA SENSITIVITY (LEVEL 2 ONLY)

This routine uses subroutine 'SLST'. Each of the General Purpose registers tested above in routine DA69 is tested again, using 44 different patterns.

ERC	Function	REG	RAC
	1- Go to subroutine 'SLST'. Parameters: starting - Output = X'06', ending - Input = X'27', data table address, table length = X'28'		
0001	2- Local store register failed		87F

DC6F - GENERAL PURPOSE REGISTER INTERACTION TEST (LEVEL 3 ONLY)

The General Purpose registers for level 1, 2, 4 and 5 that were initialized by routine DC03 are tested to check that their contents were not altered by the DCxx routines running at level 3.

Note: This routine cannot be specifically selected. It checks that the general purpose register contents set by routines DA01, DB02, DC03, and DD04 have not changed after the CCU instruction set is tested.

ERC	Function	REG	RAC
	Go to subroutine 'SGRI'. Parameters: starting - Output = X'10', ending - Input = X'07', expected data table		
0001	Interaction between level 3 and some other general registers		886

DC70 - I/O REGISTER DECODE TEST (LEVEL 3 ONLY)

This routine uses subroutine 'SIOD' to test I/O register decoding, using the General Purpose registers (each General Purpose register, from level 3, register 6 through level 1, register 7 and level 2, register 0 through level 2, register 7 is tested).

ERC	Function	REG	RAC
	Go to subroutine 'SIOD'. Parameters: starting - Output = X'0E', ending - Output = X'07'		
0001	Either an Output or an Input register decode failure occurred		87F

DC71 - GENERAL PURPOSE REGISTER DATA SENSITIVITY (LEVEL 3 ONLY)

This routine uses subroutine 'SLST'. Each of the General Purpose registers tested above in routine DB70 is tested again, using 44 different patterns.

ERC	Function	REG	RAC
	Go to subroutine 'SLST'. Parameters: starting - Output = X'0E', ending - Output X'07', data table address, table length = X'002E'		
0001	Local store register failed		87F

DD76 - GENERAL PURPOSE REGISTER INTERACTION (LEVEL 4 ONLY)

The General Purpose registers for level 1, 2, 3 and 4 that were initialized by routine DD04 are tested to check that their contents were not altered by the DDxx routines running at level 4.

Note: This routine cannot be specifically selected. It checks that the general purpose register contents set by routines DA01, DB02, DC03, and DD04 have not changed after the CCU instruction set is tested.

ERC	Function	REG	RAC
	Go to subroutine 'SGRI'. Parameters: starting - Output = X'18', ending - Input = X'07', expected data table		
0001	Interaction between level 4 and some other general registers		886

DD77 - I/O REGISTER DECODE TEST (LEVEL 4 ONLY)

This routine uses subroutine 'SIOD' to test I/O register decoding, using the General Purpose registers (each General Purpose register, from level 4, register 6 through level 1, register 7, and level 2, register 0 through level 3, register 7 is tested).

ERC	Function	REG	RAC
	Go to subroutine 'SIOD'. Parameters: starting - Output = X'16', ending - Output = X'0F'		
0001	Either an Output or an Input register decode failure occurred		87F

DD78 - GENERAL PURPOSE REGISTER DATA SENSITIVITY (LEVEL 4 ONLY)

This routine uses subroutine 'SLST'. Each of the General Purpose registers tested above in routine DB70 is tested again, using 44 different patterns.

ERC	Function	REG	RAC
	Go to subroutine 'SLST'. Parameters: starting - Output = X'16', ending - Output X'0F', data table address, table length = X'002E'		
0001	Local store register failed		87F

DA80 - LEVEL 1 TO 2 TO 5 TO 1 TEST

At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit the current level.
- Set the new program level entry address.
- Test the CCU lagging address register.

ERC	Function	REG	RAC
	1- Level 1: initialize program interrupt address (subroutine SETUP)		
	2- Set PCI L2 (Output X'7B')		
	3- Verify if other levels are pending		
0001	- In Input X'7E' (if level 1 requests not reset)		886
0002	- In Input X'7F' (level 2, 3 and 4 request, not equal, PCI L2, and interval timer L3)		
	4- Initialize program interrupt address (subroutine SETUP)		
0003	- Exit level 1 to 2		
	- Exit instruction failed to exit level 1		
0004	- Level 1 exit did not exit to level 2 but returned to level 1		
0005	- Level 1 exits to level 3 instead of level 2		
0006	- Level 1 exits to level 4 instead of level 2		
0007	- Level 1 exits to level 5 instead of level 2		
	5- Level 2: initialize program interrupt address (subroutine SETUP)		
	6- Reset PCI L2		
0008	- Verify if other levels are pending		
	7- Initialize program interrupt address (subroutine SETUP)		
	- Exit level 2 to 5		
0009	- Exit instruction failed to exit level 2		
000A	- Level 2 exit did not exit to level 5 but returned to level 2		
000B	- Level 2 exits to level 3 instead of level 5		
000C	- Level 2 exits to level 4 instead of level 5		
	9- Level 5: initialize program interrupt addresses (subroutine SETUP)		
	10- Exit Level 5 to 1		
000D	- Output instruction did not force level 1		
000E	- Level 5 exited to level 2 instead of level 1		
000F	- Level 5 exited to level 3 instead of level 1		
0010	- Level 5 exited to level 4 instead of level 1		

LEVEL 1 TO 3 TO 5 TO 1 TEST

At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit the current level.
- Set the new program level entry address.

ERC	Function	REG	RAC
	1- Level 1: initialize program interrupt address (subroutine SETUP)		
	2- Set PCI L3 (Output X'7C')		
	3- Verify if other levels are pending		
0001	- In Input X'7E' (if level 1 request not reset)		886
0002	- In Input X'7F' PCI L3 failed		
0003	Test for incorrectly set bits		
	4- Initialize program interrupt address (subroutine SETUP)		
	- Exit level 1 to 3		
0004	- Exit instruction failed to exit level 1		
0005	- The level 1 exit did not exit to level 3 but returned to level 1		
0006	- Level 1 exits to level 2 instead of level 3		
0007	- Level 1 exits to level 4 instead of level 3		
0008	- Level 1 exits to level 5 instead of level 3		
	5- Level 3: initialize program interrupt address (subroutine SETUP)		
	6- Reset PCI L3		
0009	- Verify if other levels are pending		
	7- Initialize program interrupt address (subroutine SETUP)		
	- Exit level 3 to 5		
000A	- Exit instruction failed to exit level 5		
000B	- Level 3 exits to level 1 instead of level 5		
000C	- Level 3 exits to level 2 instead of level 5		
000D	- The level 3 exit did not exit to level 5 but returned to level 3		
000E	- Level 3 exits to level 4 instead of level 5		
	9- Level 5: initialize program interrupt addresses (subroutine SETUP)		
	10- Exit Level 5 to 1		
000F	- Output instruction did not force level 1		
0010	- Level 5 exited to level 2 instead of level 1		
0011	- Level 5 exited to level 3 instead of level 1		
0012	- Level 5 exited to level 4 instead of level 1		
	12- Reset LVL1 I/O check		
0013	- I/O check reset failed		

DA83 - LEVEL 1 TO 4 TO 5 TO 1 TEST

At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit the current level.
- Set the new program level entry address.

ERC	Function	REG	RAC
	1- Level 1: initialize program interrupt address (subroutine SETUP)		
	2- Set PCI L4 (Output X'7D')		
	3- Verify if other levels are pending		
0001	- In Input X'7E' (if level 1 requests not reset)		886
0002	- In Input X'7F' Test for incorrectly set bits		
	4- Initialize program interrupt address (subroutine SETUP) - Exit level 1 to 4		
0003	- Exit instruction failed to exit level 1		
0004	- The level 1 exit did not exit to level 4 but returned to level 1		
0005	- Level 1 exits to level 2 instead of level 4		
0006	- Level 1 exits to level 3 instead of level 4		
0007	- Level 1 exits to level 5 instead of level 4		
	5- Level 4: initialize program interrupt address (subroutine SETUP)		
	6- Reset PCI L4		
0008	- Verify if other levels are pending		
	7- Initialize program interrupt address (subroutine SETUP) - Exit level 4 to 5		
0009	- Exit instruction failed to exit level 4		
000A	- Level 4 exits to level 1 instead of level 5		
000B	- Level 4 exits to level 2 instead of level 5		
000C	- Level 4 exits to level 3 instead of level 5		
000D	- The level 4 exit did not exit to level 5 but returned to level 4		
	9- Level 5: initialize program interrupt addresses (subroutine SETUP)		
	10- Exit Level 5 to 1		
000E	- Output instruction did not force level 1		
000F	- Level 5 exited to level 2 instead of level 1		
0010	- Level 5 exited to level 3 instead of level 1		
0011	- Level 5 exited to level 4 instead of level 1		
	11- Reset LVL1 I/O check		
0012	- I/O check reset failed		

- LEVEL 1 TO 5 TO 4 TO 3 TO 2 TO 1

At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit the current level.
- Set the new program level entry address.
- Test the CCU lagging address register.
- Test for an invalid Op code using the STCT instruction.

(DA84)

ERC	Function	REG	RAC
	1- Level 1: initialize pgm interrupt address (subroutine SETUP)		
	2- Verify if other levels are pending		
0001	- In Input X'7E' and X'7F'		886
	3- Initialize program interrupt address (subroutine SETUP)		
	- Exit level 1 to 5		
0002	- Exit instruction failed to exit level 1		
0003	- Level 1 exit did not exit to level 1 but returned to level 1		
0004	- Level 1 exits to level 2 instead of level 5		
0005	- Level 1 exits to level 3 instead of level 5		
0006	- Level 1 exits to level 4 instead of level 5		
	4- Level 5: Initialize pgm interrupt addresses (subroutine SETUP)		
	5- Exit level 5 to 4 (SVC L4)		
0007	- Exit instruction failed to exit level 5		
0008	- Level 5 exit did not exit to level 4 but returned to level 5		
0009	- Level 5 exits to level 1 instead of level 4		
000A	- Level 5 exits to level 2 instead of level 4		
000B	- Level 5 exits to level 3 instead of level 4		
	6- Level 4: Initialize pgm interrupt address (subroutine SETUP)		
	7- Reset SVC LVL4, then verify if other levels are pending		
000C	- In Input X'7E'		
0030	- Waiting for a 100-ms timer level 3 interruption		841
0011	- SVC LVL4 not reset		
	9- Initialize program interrupt address (subroutine SETUP)		
	- Set PCI L3, then exit level 4 to 3		
000D	- PCI L3 failed		
000E	- Level 4 exits to level 2 instead of level 3		
000F	- Level 4 exits to level 4 instead of level 3		
0010	- Level 4 exits to level 5 instead of level 3		
	10- Level 3: initialize pgm interrupt address (subroutine SETUP)		
0012	11- PCI L3 not set		
	12- Verify if other levels are pending		
0013	- Test for incorrectly set bits in Input X'7F'		
	- Reset PCI L3		
0014	- Reset PCI L3 failed		
0015	- Test for incorrectly set bits in Input X'7F'		
	13- Initialize program interrupt address (subroutine SETUP)		
	- Set PCI L2, then exit level 3 to 2		
0016	- PCI L2 failed		
0017	- Level 3 exits to level 1 instead of level 2		
0018	- Level 3 exits to level 3 instead of level 2		
0019	- Level 3 exits to level 4 instead of level 2		
001A	- Level 3 exits to level 5 instead of level 2		
	14- Level 2: initialize pgm interrupt address (subroutine SETUP)		
001B	15- Reset PCI L2 failed		
001C	- Verify if other levels are pending		
	16- Initialize program interrupt address (subroutine SETUP)		
	- Exit level 2 to 1 by invalid Op code (STCT instruction B field = R0)		
001D	- Exit failed		
001E	- The level 2 exit did not exit to level 1 but returned to level 2		
001F	- Level 2 exits to level 3 instead of level 1		
0020	- Level 2 exits to level 4 instead of level 1		
0021	- Level 2 exits to level 5 instead of level 1		
	17- Interrupt handler level 1		
0022	- Verify if invalid op code on Input X'7E' (bit 0, 4)		
	- Reset invalid op code		
0023	- Test for incorrectly set bits in Input X'7E'		
	18- Reset L1 op code check Output X'77' (bit 1, 5)		
	19- Reset all program entered latches and I/O check		
0024	20- I/O check reset failed		

- LEVEL 1 TO 2 TO 3 TO 4 TO 5 TO 1 TEST

At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit the current level.
- Set the new program level entry address.

ERC	Function	REG	RAC
	1- Level 1: Initialize program interrupt address (subroutine SETUP)		
	2- Set PCI L2 Output X'7B'		
	3- Verify if other levels are pending		
0001	- In Input X'7E' (if level 1 requests not reset)		886
0002	- Test for incorrectly set bits in Input X'7F'		
	4- Initialize program interrupt address (subroutine SETUP)		
	- Exit level 1 to 2		
0003	- Exit instruction failed to exit level 1		
0004	- Level 1 exit did not exit to level 2 but returned to level 1		
0005	- Level 1 exits to level 3 instead of level 2		
0006	- Level 1 exits to level 4 instead of level 2		
0007	- Level 1 exits to level 5 instead of level 2		
	5- Level 2: Initialize program interrupt address (subroutine SETUP)		
	6- Reset PCI L2, Set PCI L3		
	7- Verify if other levels are pending		
0008	- Reset PCI L2 failed		
0009	- Set PCI L3 failed		
000A	- Test for incorrectly set bits in Input X'7F'		
	8- Initialize program interrupt address (subroutine SETUP)		
	- Exit level 2 to 3		
000B	- Exit instruction failed		
000D	- Level 2 exit did not exit to level 3 but returned to level 2		
000E	- Level 2 exits to level 4 instead of level 3		
000F	- Level 2 exits to level 5 instead of level 3		
	9- Level 3: initialize program interrupt address (subroutine SETUP)		
	10- Reset PCI L3, Set PCI L4		
	11- Verify if other levels are pending		
0010	- Reset PCI L3 failed		
0011	- Set PCI L4 failed		
0012	- Test for incorrectly set bits in Input X'7F'		
	12- Initialize program interrupt address (subroutine SETUP)		
	- Exit level 3 to 4		
0013	- Exit instruction failed		
0014	- Level 3 exits to level 1 instead of level 4		
0015	- Level 3 exits to level 2 instead of level 4		
0016	- Level 3 exit did not exit to level 3 but returned to level 3		
0017	- Level 3 exits to level 5 instead of level 4		
	13- Level 4: Initialize program interrupt address (subroutine SETUP)		
	14- Reset PCI L4		
	15- Verify if other levels are pending		
0018	- Test for incorrectly set bits		
	16- Initialize program interrupt address (subroutine SETUP)		
	- Exit level 4 to 5		
0019	- Exit instruction failed		
001A	- Level 4 exits to level 1 instead of level 5		
001B	- Level 4 exits to level 2 instead of level 5		
001C	- Level 4 exits to level 3 instead of level 5		
001D	- Level 4 exit did not exit to level 5 but returned to level 4		
	17- Level 5: Initialize program interrupt address (subroutine SETUP)		
	- Exit level 5 to 1		
001E	- Output instruction did not force level 1		
001F	- Level 5 exits to level 2 instead of level 1		
0020	- Level 5 exits to level 3 instead of level 1		
0021	- Level 5 exits to level 4 instead of level 1		
	18- Interrupt handler LVL 1		
	- Reset I/O check		
0022	- Verify if LVL 1 pending		

STANDARD INTERRUPT HANDLERS

All routines use interrupt level 1. Only routines 80 through 85 use levels 2 through 5.

Error Reference Code Definition

```
X X X X   Error issued by CCU interrupt handlers
| | +-+-  Interrupt error sequence
| +-----  Interrupt level
+-------  Interrupt handler level (1, 2, 3, 4 or 5)
```

Output X'77'

- Reset IPL L1 (Bit 0, 0).
- Reset the CCU Hard Check (Bit 0, 1).
- Reset the Soft Check (Bit 1, 5).

ERC	Function	REG	RAC
	1- Unexpected interrupts when level 1 is running. Level 1 is interrupted by:		
1100	- Level 1		886
2100	- Level 2		
3100	- Level 3		
4100	- Level 4		
5100	- Level 5		
	2- Unexpected interrupts when level 2 is running. Level 2 is interrupted by:		
1200	- Level 1		
2200	- Level 2		
3200	- Level 3		
4200	- Level 4		
5200	- Level 5		
	3- Unexpected interrupts when level 3 is running. Level 3 is interrupted by:		
1300	- Level 1		
2300	- Level 2		
3300	- Level 3		
4300	- Level 4		
5300	- Level 5		
	4- Unexpected interrupts when level 4 is running. Level 4 is interrupted by:		
1400	- Level 1		
2400	- Level 2		
3400	- Level 3		
4400	- Level 4		
5400	- Level 5		
	5- Unexpected interrupts when level 5 is running. Level 5 is interrupted by:		
1500	- Level 1		
2500	- Level 2		
3500	- Level 3		
4500	- Level 4		
5500	- Level 5		
	6- Interrupt handler LVL 1 from level 5 - Verify if PGM level 5 interrupted (Input X'79' byte 1, bit 3)		
1501	- Verify if L5 I/O error on Input X'7E' bit (0, 3) - Test CCU lagging address register, load expected address in R3 - Input (X'74'), LAR in R1, and compare input address with expected address in R3		
0011	- Test Z latch - Reset L5 I/O Output X'77' bit (1,5) - Restore program interrupt address		

ROUTINE SLST: GENERAL PURPOSE REGISTERS TEST

This routine loads a 22 bit pattern (using a pattern table) successively into each General Purpose register not used by the current level (using an 'Out' instruction). It then reads it back (using an 'In' instruction) and compares the written and read patterns.

ERC	Function	REG	RAC
	1- Parameters: Output to start (local store address), Output to end (local store address), address of pattern table, table length, stop if error 2- Load data table address and save it. Load first data pattern, load iteration count and save it. Load first Output instruction and modify it to an Input Instruction. 3- Loop while swapping Output instruction and Input instructions to load successively each data pattern into each general purpose register not used by the current level and compare the write pattern with the read pattern If compare is not OK, verify if possible valid level 3 IAR (timer) Stop loop if expected last Output reached, load current data pointer, load next data branch and count (current data pointer) to 1		

SUBROUTINE SIOD: IN/OUT REGISTER DECODE

Load the Output instruction into each General Purpose register not used by the current level. Read each General Purpose register and compare it with the expected result (calculated in this second phase).

ERC	Function	REG	RAC
	1- Parameters: Output to start (local store address), Output to end (local store address), stop if error 2- Load and store Output instruction to modify it 3- Execute Output instruction, update Output instruction external register value, loop on 2 to load each General Purpose register not used by the current LVL 4- Load first Output instruction and modify it to an Input instruction 5- Execute the Input instruction and compare with expected result (calculated in this second phase) If compare not OK, verify if possible valid level 3 IAR (timer). Stop loop if last Output reached		

SUBROUTINE SRGI: REGISTER INTERACTION TEST

This routine compares the contents of the General Purpose registers, initialized before running the routines for each level, with an expected data table.

ERC	Function	REG	RAC
	1- Parameters: Output to start (local store address), Input to end (local store address), table of expected data, stop if error 2- Clear registers 1 through 3, load compare table address, load first Input instruction 3- Modify LS address in Input instruction, load next instruction, load next compare data, update table address 4- Execute modified Input, compare (did previous tests modify registers), return to error stop if error after verifying possible level 3 IAR (timer) 5- Stop loop if last Input reached		

SUBROUTINE SETUP: INITIALIZE LEVEL EXIT, RESET INTERRUPT MASK

Depending on table entries, this subroutine loads a link address to the level interrupt handler, sets mask or unmask field, and resets the interrupt level mask.

ERC	Function	REG	RAC
	1- Parameters: Flags, as follows: Link address to Level 1 Link address to Level 2 Link address to Level 3 Link address to Level 4 Link address to Level 5 Mask field Unmask field Interrupt reset mask 2- Load flag field and complement it, clear register 3- Update pointer to next HW parameter Update pointer to next flag field Link to test next flag 4- Modify level 1 address 5- Modify level 2 address 6- Modify level 3 address 7- Modify level 4 address 8- Modify level 4 address (Load level 5 IAR) 9- Set new mask (Out X'7E') 10- Set new unmask (Out X'7F') 11- Reset interrupt level		

SUBROUTINE SBXT: BYTE X TEST

This routine shifts byte X into byte 0 to test it.

ERC	Function	REG	RAC
	1- Shift byte X into byte 0 using six LOR instructions and clear register (1) 2- Load expected data and compare		

Each routine of IFT E communicates with subroutines located in CCU storage, packaged in a subordinate section. The subordinate section returns control to the IFT via an Out X'70' instruction. To start program execution at level 1, the routine resets program stop. The routine waits for a high level interrupt from the CCU. If no interrupt occurs, this is reported via the DCM.

EA01 - STORAGE TEST ADDRESSING (WHOLE STORAGE EXCEPT FIRST 200 BYTES)

The subordinate section loaded at CCU address 0 performs tests from the first fullword next to the end of the subordinate section to the end of installed storage. Three patterns are used: X'55555555', X'AAAAAAAA' and X'31313131'. Each pattern is stored at a fullword address and loaded from the same address and both patterns are compared. Any error is reported via an Out X'70' instruction (high level interrupt) to the MOSS.

ERC	Function	Error Description	RAC	Comments
0700	Store and load a pattern to a particular fullword	Loaded pattern not equal to stored pattern	8E1 to 8E8 789 to 78C	Last digit of RAC gives failing storage card
0701	Interrupt to MOSS expected at the end of test	No interrupt	860	

EA02 - STORAGE TEST FORWARD/BACKWARD DISTURBANCE

(Whole Storage Except First 200 Bytes)

The subordinate section loaded at CCU address 0 tests the main storage from the first fullword next to the end of subordinate section up to the end of installed storage. The whole tested storage is written with X'0'. A selected word is then written with its own address. The whole storage is then read to check that only one particular fullword was modified. The test is restarted for each fullword of tested storage.

ERC	Function	Error Description	RAC	Comments
0700	Store and load a pattern of X'0' to the whole tested storage	Loaded pattern not equal to the stored pattern.	8E1 to 8E8	Last digit of RAC gives failing storage card
0701	Interrupt to MOSS expected at the end of test	No interrupt	860	

EB01 - STORAGE TEST ADDRESSING (FIRST 128K)

The subordinate section loaded at address X'20000' (second block of 128K bytes) tests the first block of 128K bytes. Each fullword is stored and loaded from with three patterns:

- X'55555555', X'AAAAAAAA', X'31313131'.

The stored and loaded patterns are compared.

ERC	Function	Error Description	RAC	Comments
0700	Store and load a pattern to a particular fullword	Loaded pattern not equal to stored pattern	8E1	
0701	Interrupt to MOSS expected at the end of test	No interrupt	860	

EB02 - STORAGE TEST, FORWARD/BACKWARD DISTURBANCE (FIRST 128K)

The subordinate section loaded at address X'20000' (second block of 128K bytes) tests the first block of 128K bytes. The whole tested area is written with X'0'. A selected word is then written with its own address. The whole tested area is read to check that only one particular fullword was modified. The test is restarted for each fullword of the tested area.

ERC	Function	Error Description	RAC	Comments
0700	Store and load a pattern of X'0' to the first card.	Loaded pattern not equal to stored pattern	8E1	
0701	Interrupt to MOSS expected at the end of test	No interrupt	860	

- **BRANCH TRACE (LEVEL 1) TEST**

1. Initialize the branch trace mechanism and the branch trace buffer.
2. Load the exerciser code into CCU storage.
3. Start the CCU.
4. When the branch trace buffer is full, the CCU stops with a low level interrupt to the MOSS.
5. Read the branch trace buffer and compare the records in the branch trace buffer.

Note: the exerciser code runs at level 1.

ERC	Function	Error Description	RAC	Comments
0700	Test for expected low level interrupt	No low or high level interrupt occurred	895	
0701	Compare branch trace buffer with the internal table	An error occurred in the branch trace buffer	89E	ERR BIT field on the screen displays the first 4 bytes of the entry in error

FA02 - BRANCH TRACE TEST (LEVELS 1, 2, 4, AND 5)

1. Initialize the branch trace mechanism (to trace 4 levels) and the branch trace buffer.
2. Load the exerciser code for 4 levels.
3. Start the CCU at level 1.
4. Exit to level 2, 4, or 5.
5. When the branch trace buffer is full, the CCU stops with a low level interrupt to the MOSS.
6. Read the branch trace buffer and compare records in the branch trace buffer.

ERC	Function	Error Description	RAC	Comments
0701	Read CCU-to-MOSS Status C register. Check for value X'A2' (Branch Trace interrupt, CCU Stop due to BT full, Program Stop)	CCU-to-MOSS Status C register does not contain X'A2', or a high level interrupt occurred	895	
0702	Compare branch trace buffer with the internal table	An error occurred in the branch trace buffer	89E	ERR BIT field on the screen displays the first 4 bytes of the entry in error

FA03 - SINGLE ADDRESS COMPARE ON LOAD INSTRUCTION TEST

1. Initialize the address compare mechanism in order to stop on a load instruction.
2. Load the exerciser code into the CCU.
3. Start the CCU.
4. Test the low level interrupt due to a successful address compare.
5. Verify the contents of the SAR for a correct load address.

ERC	Function	Error Description	RAC	Comments
0700	Test that a high-level interrupt is absent, and a low-level interrupt is present.	Either high-level interrupt did occur or the low-level interrupt did not occur.	895	
0701	Test SAR and IAR	SAR or IAR contains bad address value.	89E	ERR BIT field on the screen displays value of SAR

F - SINGLE ADDRESS COMPARE ON STORE INSTRUCTION TEST

1. Initialize the address compare mechanism in order to stop on a store instruction.
2. Load the exerciser code into the CCU.
3. Start the CCU.
4. Test the low level interrupt due to a successful address compare.
5. Verify the contents of the SAR for a correct store address.

ERC	Function	Error Description	RAC	Comments
0700	Test that a high-level interrupt is absent, and a low-level interrupt is present.	Either high-level interrupt did occur or the low-level interrupt did not occur.	895	
0701	Test SAR (should contain the correct store address)	SAR contains a different address value	89E	ERR BIT field on the screen displays value of SAR

FA05 - DOUBLE ADDRESS COMPARE ON LOAD INSTRUCTION TEST

1. Initialize the double address compare mechanism on load instruction and instruction fetch.
2. Load the exerciser code into the CCU.
3. Start the CCU.
4. Test the low level interrupt due to a successful address compare.
5. Verify the IAR and the SAR for the instruction address and the address of the operand.

ERC	Function	Error Description	RAC	Comments
0700	Test that a high-level interrupt is absent, and a low-level interrupt is present.	Either high-level interrupt did occur or the low-level interrupt did not occur.	895	
0701	Test IAR and SAR, (should contain the address of the instruction and the address of the operand respectively)	Either SAR or IAR contains a different address	89E	ERR BIT field on the screen displays value of SAR

FA06 - DOUBLE ADDRESS COMPARE ON STORE INSTRUCTION TEST

1. Initialize the double address compare mechanism on store instruction and instruction fetch.
2. Load the exerciser code into the CCU.
3. Start the CCU.
4. Test the low level interrupt due to a successful address compare.
5. Verify the IAR and the SAR for the instruction address and the address of the operand.

ERC	Function	Error Description	RAC	Comments
0700	Test that a high-level interrupt is absent, and a low-level interrupt is present.	Either high-level interrupt did occur or the low-level interrupt did not occur.	895	
0701	Test IAR and SAR, (should contain the address of the instruction and the address of the operand respectively)	Either SAR or IAR contains a different address	89E	ERR BIT field on the screen displays value of SAR

FA07 - TWO SINGLE ADDRESS COMPARE ON INSTRUCTION FETCH TEST

1. Initialize the two single address compare mechanism on instruction fetch without CCU stop.
2. Load the exerciser code into the CCU.
3. Start the CCU.
4. Test the high level interrupt due to an Out X'70' at the end of the exerciser.
5. Verify the CCU-to-MOSS Status C register and the IAR.

ERC	Function	Error Description	RAC	Comments
0700	Verify high level interrupt caused by the Out X'70' at the end of the exerciser	No high level interrupt occurred	895	
0701	Verify in the CCU-to-MOSS Status C register that an address compare interrupt and 2 single address compare on address 2 bits are set. Verify that the IAR contains the address of the second address compare.	Either CCU-to-MOSS Status C register is not correctly set or IAR contains bad address.	89E	ERR BIT field on the screen displays value of IAR

CHAPTER 3. IOCB AND RDV DIAGNOSTICS

Table of Contents for Chapter 3

CHAPTER 3. IOCB AND RDV DIAGNOSTICS

The IOC bus diagnostic group is divided into two IFTs that test:

- RDV (IFT I)
- IOC Bus (IFT J)

REQUIREMENTS

These IFTs run under the control of the diagnostic control monitor (DCM) in MOSS, and the communication processor (CP) in CCU storage.

Unless directed by MIM part 2 to do otherwise, you must ensure that the CCU IFTs work properly before running the RDV IFT. If not, the results given by the RDV IFT may be of no value, or misleading.

SELECTION

For running offline diagnostics, see Chapter 1.

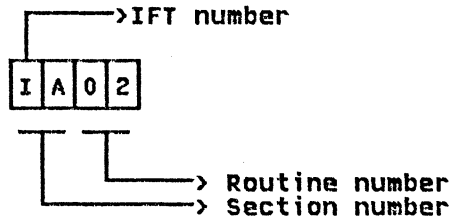
DIAG==>_

3	IOCB group selected
X	Specific IFT X in this group
XY	Specific section y in IFT X
XYZZ	Specific routine ZZ in section XY

For specific section and routine selection, see routine lists on following pages, and in Chapter 1.

Move the cursor from its initial position (DIAG==>) to the next after each parameter is entered. To skip a parameter entry, press the --> key.

To interpret correctly the results of a selected section or routine, make sure the preceding IFTs, sections, and routines in the group are running without error. The routine identification contains the IFT number, the section number, and the routine number as follows:



ADP#==>_ (not applicable)

LINE==>_ (not applicable)

OPT==> N For option display and description, see Chapter 1.

SELECTION RESTRICTIONS

EXPLICIT SELECTION

Selection of IFT J or section number (JA, JB or JC) is possible.

Selection of routine JAxx should not be attempted because it will create false error reporting.

Note: Routines of Section JA run once per scanner.

CYCLE ON REQUEST

Cycle on request for IFT J or section number (JA, JB, or JC) is possible.

Cycle on request for routine numbers (JAxx) is not possible.

Cycle on request for routine number (JBB0) is possible.

REPEAT OPTION

Repeat request for IFT J or section number (JA) is not possible.

Repeat request for section (JB) or for routine number (JBB0) is possible.

Repeat request for section (JC) or for routine number (JC01 or JC02) is possible.

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK

D:DIAGNOSTICS                              SP:CCU STOP    Q:DATE/TIME
                              U:UTILITY PGM ST:CCU START   T:TERMINATE
E:ERROR LOG                                RS: CCU RESET

DIAG   |ADP# |LINE |
1 ALL
2 CCU
3 IOCB
4 CA   |1->6 |     |
5 TSS  |1->16|0->31|
6 OLT  |1-> 6|     |
7 TRSS|1->16|1->4  |

       AND                                      DIAG - RUN INIT

OPT = Y IF MODIFY
OPTION REQUIRED
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> 3     ADP#==>     LINE==>     OPT==> N
===>
```

On the above screen, the IOC bus group is selected, without option modification.

Press SEND to execute the request.

Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

LOOP ON ERROR

Loop on error is not allowed when an error occurs in routines JA80 or JA81.

MANUAL INTERVENTION ROUTINE

There is no manual intervention routine in the IOC bus diagnostic group.

BRANCH TRACE

When running IFT J, the branch trace of the CCU functions is not allowed.

IOC BUS DIAGNOSTIC GROUP RUNNING TIME

When the diagnostic request is 3, the total running time is:

1 min 37 sec + (15 sec *X) + (6 sec *Y) + (10 sec *Z)

Where:

- X is the total number of CLABs + LABs installed,
- Y is the total number of CLABs + CABs installed.
- Z is the total number of LAB type C installed (maximum 2).

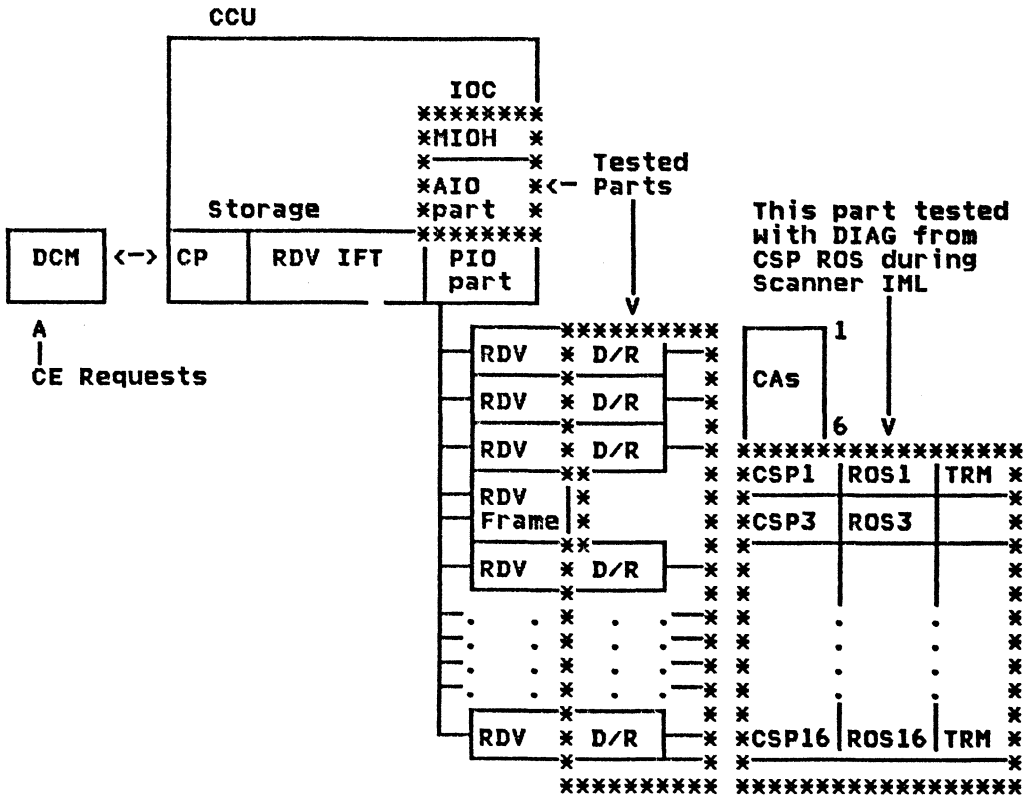


Figure 3-1. RDV IFT

RDV IFT I

This IFT exercises the IOC bus circuits from the CCU and the MOSS.

The routines that compose the RDV IFT check every function on every RDV card. The drivers and receivers that connect an RDV to its adapter (channel adapter or CSP) are disabled during routine IA01 to prevent reaction from the adapter on the RDV card. The other routines of the RDV IFT are run with the drivers and receivers enabled.

ROUTINE IA01

This routine is entered once only to test the IOC circuit used by the following routines. If an error is detected, it calls for manual intervention by the CE to determine the failing RDV.

ROUTINES IA02 TO IA71

A routine tests every RDV in sequence before the next routine is called. RDVs are tested if present in the 3725 CDF information given to the DCM.

RUNNING TIME

Run Init IA	1 min 10 sec (see note) 14 sec

Note: The run initialization takes only 13 sec when the command processor has already been loaded by a previous IOC bus, or by a CA run request.

MESSAGES

If a failure is detected by the first routine (IA01) of the IOC bus group of IFTs, a message asks the CE to perform a specific action. These messages are:

DISC. RDVs 2 3 4 AND 10 IF ANY THEN PRESS SEND

All these redrives are on the primary bus.

Although DISC. means disconnect, the intent of this message is to have the CE install board disable jumpers from pin D11 to D08 of each RDV listed.

Refer to 'Redrive State Definitions' on page 11-090 of the 3725/3726 MIM. For RDV card locations, refer to page 4-070 of the 3725/3726 MIM. Because RDVs 4 and 10 are located in the second frame, they are not present if this frame is absent.

RECONNECT RDV 2 THEN PRESS SEND

DISC. RDV 1 AND REC. RDV 2 THEN PRESS SEND

RECONNECT RDV 3 THEN PRESS SEND

RECONNECT RDV 10 THEN PRESS SEND

RECONNECT RDV 4 THEN PRESS SEND

The following messages are displayed only if the fault is on the secondary bus:

DISC. RDVS 5 6 7 8 AND 9 IF ANY THEN PRESS SEND

If a RDV card is not installed, do not perform the requested action for that redrive.

DISC. RDVS 6 7 8 AND 9 IF ANY THEN PRESS SEND

If an RDV card is not installed, do not perform the requested action for that redrive.

DISC. RDVS 7 8 AND 9 IF ANY THEN PRESS SEND

DISC. RDV 5 AND REC. RDV 6 THEN PRESS SEND

DISC. RDV 6 AND REC. RDV 7 THEN PRESS SEND

RECONNECT RDV n THEN PRESS SEND

where n is a redrive of the secondary bus

Note: When all these actions have been taken, the IFT indicates the failing RDV via an RAC displayed on the screen. Do not forget to remove the jumpers or reconnect the cables before running the IOC Bus diagnostics, otherwise you will have problems due to unplugged or disconnected RDVs.

IOC BUS IFT J

This IFT sends an IOC command reset on the IOC bus. This starts the diagnostic stored in the ROS of every scanner. When its internal testing is over, the scanner enters a loop, waiting for an IFT request to resume (PIO operation).

ROUTINES JA80 AND JA81

These routines test the scanner part not tested with the ROS diagnostic, the DFL5 card, and the IOC bus.

ROUTINE JBB0

This routine tests the connection between CCU and every channel adapter.

ROUTINE JC01

This routine tests the ability to read and write to the diagnostic register.

ROUTINE JC02

This routine tests the ability to force a level 2 interrupt and check the result.

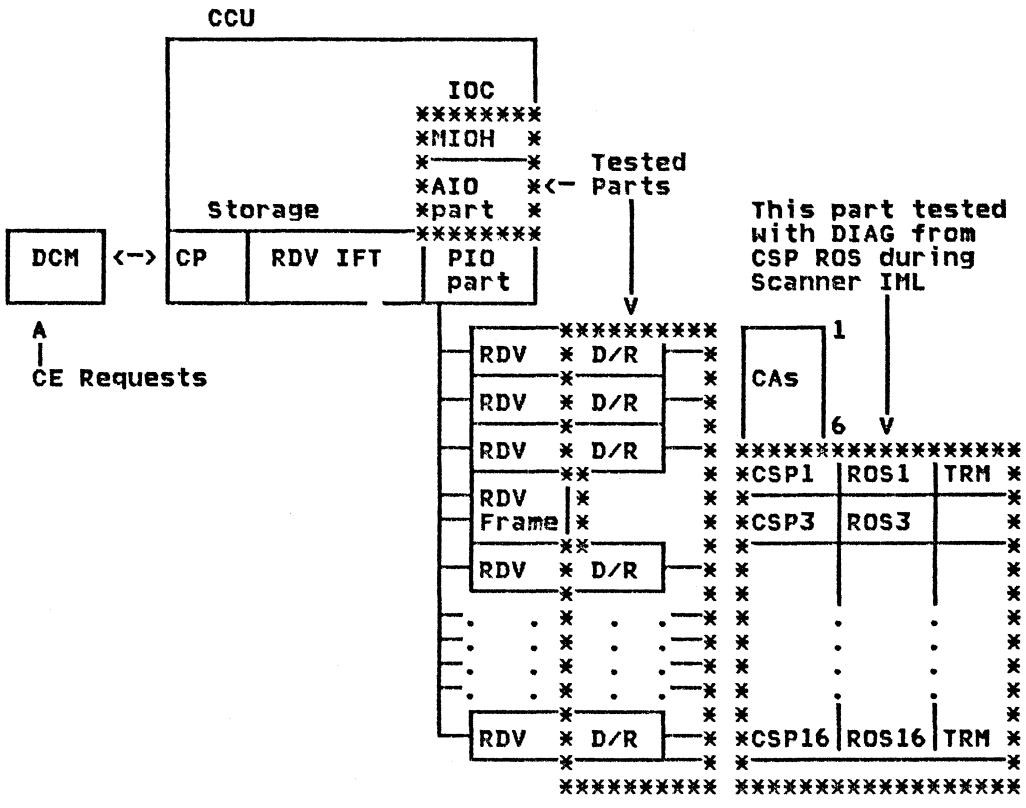


Figure 3-2. IOC Bus IFT

RUNNING TIME

Run Init	1 min 10 sec (see note)
JA	15 times the total number of CLABs and LABs installed (time in seconds)
JB	6 times the total number of CLABs and CAB installed (time in seconds)
JC	10 times the total number of LAB type C installed (time in seconds)

Note: The run initialization takes only 30 sec when the command processor has already been loaded by a previous IOC bus, or by a CA run request.

MESSAGES

JA80

If a CSP and a TRM are installed on the same LAB (type C), and an error is detected, the second adapter is checked for a TRM. If a TRM is present, the message "LABC" is displayed in the ADDIT INFO field. The error may be caused by either the CSP or the TRA.

ROUTINES DESCRIPTION

IA01 - PRIMARY TEST OF IOC BUS USING RDVS

The following set of hexadecimal patterns: X'FF', X'B7', X'DC', X'02', X'69', X'F0', X'CC', X'33', X'55', X'AA', and X'00', is written and read back using the PIO to each installed redrive. The error register is used for the PIO.

The above test is done twice. The first iteration is done with a disabled redrive and the second with an enabled redrive. If an error is found then a correlation test is dispatched for isolation purposes.

ERC	Function	Error Description	RAC	Comments
FFFF	Second iteration run	An error occurred while the RDV(s) were enabled	602	Run IFT J
7002 to 700D	Isolate the failing RDV	The same set of patterns is used for isolation. RDVn failed.	602	
7000	Correlation done via the disconnect procedure	Error found on every RDV	601	
7030 to 703B	Correlation done via the disconnect procedure	Error found on RDV2	602	
7040 to 704B	Correlation done via the disconnect procedure	Error found on RDV3	602	
7050 to 705B	Correlation done via the disconnect procedure	Error found on RDV10	602	
7060 to 706B	Isolation of the secondary bus via the disconnect procedure	Error found on RDV4	60C	
7070 to 70A0	Secondary bus isolation	Error found on RDV4 or RDVn (n: RDV of secondary bus	60C	
70EE	Correlation done on secondary bus and error found on all RDVs	Error occurs on all secondary RDVs	60D	ERR BIT field gives bits on error
70FF	Correlation done and error found on one redrive	Error found on RDVn	602	ERR BIT field gives bits on error
7001	Correlation dispatched due to an error found; however only RDV1 is installed in the system	Error found for RDV1	60B	ERR BIT field gives bits on error

IA02 - ADDRESS DECODE PHASE 1

This routine addresses a non-installed redrive (not present in the CDF). The address is verified to determine if another one has replied (installed acknowledged). The test is done for every non-installed RDV.

Note: This routine is bypassed if the 3725 is fully equipped (all RDVs installed).

ERC	Function	Error Description	RAC	Comments
70C0	Write a data pattern into error reg of a non-installed RDV	No timeout occurred while addressing a non-installed RDV	601	ERR BIT field gives bits on error
70C1	Level 1 interrupt processing	An interrupt L1 occurred, but not for a timeout.	601	

IA03 - REDRIVE ERROR REGISTER TEST

A set of different patterns is sent via a PIO Write and read back via a PIO Read to every installed redrive using the error register.

A verification is made when the Read is completed; the result of the Read should be X'aadd' (where X'aa' is the redrive address and X'dd' is the data pattern expected in byte 0).

ERC	Function	Error Description	RAC	Comments
	Write a pattern and check the data pattern received and the redrive address	Mismatch for data pattern and redrive address	602	ERR BIT field gives bits on error
70C6	X'FF00' pattern			
70C7	X'B700' pattern			
70C8	X'DC00' pattern			
70C9	X'0200' pattern			
70CA	X'6900' pattern			
70CB	X'F000' pattern			
70CC	X'CC00' pattern			
70CD	X'3300' pattern			
70CE	X'5500' pattern			
70CF	X'AA00' pattern			
70D0	X'0000' pattern			
70D1	Level 1 interrupt processing	A level 1 interrupt occurred on a PIO:		
		- caused by a timeout	602	
		- caused by another reason	601	

IA04 - ADDRESS DECODE PHASE 2

1. The error register of every installed redrive is initialized to all '0's via a PIO Write Specific.
2. Starting with the first installed redrive, the data pattern X'5500' is written into the RDV error register via a PIO Write Specific command.
3. A Read Specific command is then performed to check first the data pattern and second the RDV address.
4. The second iteration writes all '0's into the RDV error register. All the registers of the addressed redrive should contain all '0's.
5. Steps 2 through 4 are repeated for each redrive.

ERC	Function	Error Description	RAC	Comments
70E0	Check data pattern X'55'	Mismatch in the data pattern	602	ERR BIT field gives bits on error
70E1	Check the RDV address after the Read Error Register	Returned RDV address is wrong	602	ERR BIT field gives bits on error
70E2	Check data pattern X'00'	Error reg of addressed RDV is not empty	602	ERR BIT field gives bif on error
70E3	Check the RDV address after the Read Error Reg	Returned RDV address is wrong	602	ERR BIT field gives wrong address
70E4	Level 1 interrupt processing	A level 1 interrupt occurred on a PIO: - caused by a timeout - caused by another reason	 602 601	

IA05 - AUTOMATIC RESET CHECKING

1. The error register of each installed RDV is written with the pattern X'FF00' via a PIO Write Specific command; this register is then read back to verify the written pattern.
2. A data pattern of X'A500' is next sent via a PIO to an adapter group other than the redrives.
3. The routine checks that each RDV error register has been reset.

ERC	Function	Error Description	RAC	Comments
70E7	Check X'FF00' data pattern sent and received.	Mismatch in the data pattern	602	ERR BIT field gives bits on error
70E8	Check that the RDV error reg has been reset	RDV error reg is not reset	602	ERR BIT field gives bits on error
70EA	Level 1 processing for Write/Read pattern X'FF00'	Level 1 interrupt occurred on a Write/Read PIO. The cause of the interrupt is a timeout. Level 1 interrupt occurred on a Write/Read PIO. The cause of the interrupt is not a timeout.	602 601	
70EB	Expected level 1 interrupt when addressing a non-implemented adapter	Level 1 interrupt occurred but it is not a timeout.	601	
70EC	Expected level 1 interrupt when addressing a non-implemented adapter	No level 1 interrupt occurred.	602	

IA06 - WRITE ERROR REGISTER BROADCAST TEST

1. This routine writes the pattern X'8800' into each RDV error register via a PIO Write Broadcast command.
2. Each RDV error register is then read back via a PIO Read Specific command.
3. Checking is done for the data pattern received and during level 1 processing.

ERC	Function	Error Description	RAC	Comments
70F1	Check if received data pattern is X'8800'	Data pattern does not match	602	ERR BIT field gives bits on error
70F0	Level 1 interrupt processing	Level 1 interrupt occurred on a PIO: - caused by a timeout - caused by another reason	 602 601	

IA07 - DIAGNOSTIC REGISTER WRITE COMMAND TEST

1. This routine sends a set of patterns via a PIO Write Diagnostic Register Specific command to each installed RDV.
2. The routine then reads back the diagnostic register of every installed RDV and checks the result.

ERC	Function	Error Description	RAC	Comments
70F5	Check data pattern X'FF'	Data pattern not X'FF'	602	(note)
70F6	Check data pattern X'B7'	Data pattern not X'B7'	602	(note)
70F7	Check data pattern X'DC'	Data pattern not X'DC'	602	(note)
70F8	Check data pattern X'02'	Data pattern not X'02'	602	(note)
70F9	Check data pattern X'69'	Data pattern not X'69'	602	(note)
70FA	Check data pattern X'F0'	Data pattern not X'F0'	602	(note)
70FB	Check data pattern X'CC'	Data pattern not X'CC'	602	(note)
70FC	Check data pattern X'33'	Data pattern not X'33'	602	(note)
70FD	Check data pattern X'55'	Data pattern not X'55'	602	(note)
70FE	Check data pattern X'AA'	Data pattern not X'AA'	602	(note)
70FF	Check data pattern X'00'	Data pattern not X'00'	602	(note)
70F5 to 70FF	Level 1 interrupt processing	Level 1 interrupt occurred and it is a timeout	602	
		Level 1 interrupt occurred and it is not a timeout	601	
Note: 'ERR BIT' field gives bits on error				

IA08 - DIAGNOSTIC REGISTER WRITE BROADCAST TEST

1. This routine writes the pattern X'FFFF' into each RDV diagnostic register using a Write Broadcast command.
2. Each RDV diagnostic register is then read back using a PIO Read Specific command and the received data pattern is checked.
3. The reset is done by writing zeros via a Write Broadcast command, and the reset is verified.

ERC	Function	Error Description	RAC	Comments
7103	Read data pattern sent via the broadcast command	Mismatch in the data	602	ERR BIT field gives bits on error
7104	Verify that the diagnostic reg has been reset	Diagnostic reg is non-zero	602	ERR BIT field gives bits on error
7105	Level 1 processing for all PIO commands	Level 1 interrupt occurred and it is a timeout	602	
		Level 1 interrupt occurred and it is not a timeout	601	

IA10 - CYCLE STEAL LINE (PHASE 1) TEST

This routine tests the first part of the daisy chain (CSR and CSG). The diagnostic register provides the means to raise these tags and to propagate the signals from one RDV to the next on the chain. This phase checks that if there is no CSR pending, there is no CSG at the output of each installed RDV.

1. Each RDV diagnostic register is set to zeros via a PIO Write Specific command.

2. The state of the following signals: CSG-H-O, CSG-H-S, CSG-L-O, CSG-L-S is verified.

ERC	Function	Error Description	RAC	Comments
7106	Check for 'Cycle Steal Grant' signal (CSG-H)	The state of this signal is up		
		- If RDV1	602	
		- If not RDV1	604	
7107	Check for 'Cycle Steal Grant' signal (CSG-L)	The state of this signal is up		
		- If RDV1	602	
		- If not RDV1	604	
7108	Level 1 interrupt Processing	Interrupt level 1 occurred		
		- caused by a timeout	602	
		- caused by another reason	601	

IA11 - CYCLE STEAL PROPAGATION (PHASE 2) TEST

This routine tests the propagation of the CSG-L-I from one RDV to the next, using the diagnostic register functions.

1. For each installed redrive, CSG-L-I is set on and CSR-L-I is set off.

2. At the output of the redrive, a check is made that CSG-L-O is on and that CSG-L-S is off.

ERC	Function	Error Description	RAC	Comments
7109	Perform a Diagnostic Read for each RDV and check: CSG-L-O is on CSG-L-S is off	CSG-L-O was found off		
		- If RDV1	604	
		- If RDV4	605	
		- If neither RDV1 nor RDV4	602	
710A	Perform a Diagnostic Read for each RDV and check: CSG-H-O is on CSG-H-S is off	CSG-H-O was found off		
		- If RDV1	604	
		- If RDV4	605	
		- If neither RDV1 nor RDV4	602	
710C	Level 1 interrupt processing	Level 1 interrupt occurred		
		- caused by a timeout	602	
		- caused by another reason	601	

IA12 - CYCLE STEAL PROPAGATION (PHASE 3) TEST

This routine first sets the CSR-L-I bit (byte 1, bit 3) in the diagnostic register and then resets it; it then checks the result in the diagnostic register (byte 0, bit 3 should be on and all other bits off).

The routine next sets the CSR-H-I bit (byte 1, bit 2) in the diagnostic register, after which it checks that all the bits of the diagnostic register

It then writes an X'002C' pattern; here also, all bits of the diagnostic register should be off.

ERC	Function	Error Description	RAC	Comments
710D	Check the result of diagnostic register when writing CSR-L-I Expected result: X'1000'	Pattern read from the diagnostic register is not the one expected	602	
710E and 710F	Check the result of diagnostic register when writing CSR-H-I Expected result: X'0000'	The diagnostic register is not empty	602	
7110	Level 1 interrupt Processing	Level 1 interrupt occurred on PIO to diagnostic reg: - caused by a timeout - caused by another reason	 602 601	

IA13 - CYCLE STEAL PROPAGATION (PHASE 4) TEST

This routine tests the active function of 'Cycle Steal Low' and 'Cycle Steal High'. For each installed RDV the following patterns are sent (the table below also shows the expected pattern):

Pattern sent	Pattern expected
X'0050'	X'5010'
X'00A0'	X'0020'
X'00A4'	X'0080'
X'00AC'	X'0020'

ERC	Function	Error Description	RAC	Comments
7114	Write the diagnostic reg with the data pattern X'0050' CSG/L-I and CSR/L-I	The result in the diagnostic reg is not the one expected X'0010'	602	
711B	Check the Cycle Steal High with data pattern X'00A4'	The result in the diagnostic reg is not the one expected X'0080'	605	
7117	Check the Cycle Steal High with data patterns X'00A0' and X'00AC'	The result in the diagnostic reg is not the one expected X'0020'	602	
7115	Level 1 interrupt processing for the first pattern	Level 1 interrupt occurred - caused by a timeout - caused by another reason	 602 601	
7116	Level 1 interrupt processing for the three other patterns	Level 1 interrupt occurred - caused by a timeout - caused by another reason	 602 601	

IA14 - CYCLE STEAL PROPAGATION (PHASE 5) TEST

This routine tests the propagation of 'Cycle Steal Grant' (L/I, L/O, L/S) from one redrive to the next.

1. On the first RDV, CSG-L/I is set on; CSG-L/O should be on at all RDVs.

2. On the secondary bus, do the following:

   Write zero into RDV#4; CSG-L/O (byte 1, bit 1) should be on.

   Write CSR-L/I (byte 1, bit 3); CSG-L/S (byte 1, bit 4) should be on.

ERC	Function	Error Description	RAC	Comments
7126	Check the distribution of 'Cycle Steal Grant L/O' on all installed redrives	The signal CSG is not found as expected. - If RDV#1 - If not RDV#1	 602 607	
7128	Check the secondary bus. When CSR-L/I is set CSG-L/S should be found for every secondary redrive.	The signal CSG-L/S is not found as expected	608	
7127	Level 1 interrupt processing for the first step.	Level 1 interrupt occurred - caused by a timeout - caused by another reason	 602 601	
712A	Level 1 interrupt processing for the second step	Level 1 interrupt occurred: - caused by a timeout - caused by another reason		

IA15 - CYCLE STEAL PROPAGATION (PHASE 6) TEST

This routine tests the propagation of the signals 'Cycle Steal Grant High Out' (CSG-H/O) and 'Cycle Steal Grant Secondary' (CSG-S) from one redrive to the next.

1. Test CSG-H/O. Set CSG-H/I on the first RDV and check for CSG-H/O on all redrives.

2. Test the priority bus. Check CSG-H/O, then write CSR-H/I and CSR-PRY/D. The output signal from each RDV should be CSG-H/S.

3. Test for 'Cycle Steal Request' on the secondary bus.

ERC	Function	Error Description	RAC	Comments
7130	On RDV1 Write Diagnostic Reg with CSG-H/I (1.0) and check for each RDV CSG-H/O	CSG-H/O is not passed to each RDV. - If RDV1 - Otherwise	 602 607	
7132	For each RDV, do the following: - Diagnostic Read gives CSG-H/O - Diagnostic Write (1.2) gives CSG-H/O - Diagnostic Write with bits (1.2, 1.4) gives CSG-H/S	Error found for CSG-H/O Error found for CSG-H/S	602 607	
7134	Check the secondary bus: For each secondary RDV do: Diagnostic Write with CSR-H/I (1.2) should gives CSG-H/S (1.3)	The CSG-H/S was not found	608	
7131	Level 1 interrupt processing for the first step.	Level 1 interrupt occurred - caused by a timeout - caused by another reason	 602 601	
7133	Level 1 interrupt processing for the second step.	Level 1 interrupt occurred - caused by a timeout - caused by another reason	 602 601	
7135	Level 1 interrupt processing for the third step.	Level 1 interrupt occurred - caused by a timeout - caused by another reason	 602 601	

IA20 - POLL FUNCTION OF RDV (PHASE 1) TEST

This routine tests the propagation of the signal 'Allow Poll Response'. The CCU (IOC) forces APR-I on the first RDV which in turn propagates it to the next installed RDV. The diagnostic register is used to test the Poll function. This routine uses a table having 4 entries.

Each entry contains:

- the data to be written
- the expected data
- the L1 interrupt expected
- the RAC.

ERC	Function	Error Description	RAC	Comments
713A	Get the first entry of table	Error occurred when testing the APR-0 function	602	ERR BIT field gives bits on error
713B	Write the L1 remember latch (byte 1, bit 6). When polled, the first RDV should respond and send its own address. The diagnostic reg must have APR-0 (1.5)	Error occurred when testing the poll command	604	ERR BIT field gives bits on error
713C 713D	Second entry is used to test the level 1 pending latch and the APR-S	Error occurred. The APR-S is off	604	ERR BIT field gives bits on error
		Error occurred for the Poll command.	604	ERR BIT field gives bits on error
713E 713F	Third entry is used to check that when no level 1 is pending, timeout occurs on a Poll and APR-0 in the diagnostic reg (1.5) is on.	No timeout occurred when it was expected	602	ERR BIT field gives bits on error
		APR-0 is not on	602	ERR BIT field gives bits on error
7140 7141	Fourth Entry is used to set up L1 Remember latch and L1 pending in the diagnostic reg (1.6, 1.7). Expect to find APR-S and APR-0 off.	When issuing the Poll, time-out occurred	602	ERR BIT field gives bits on error
		APR-S and APR-0 are not off	602	ERR BIT field gives bits on error
7142	Level 1 interrupt processing	Level 1 interrupt occurred on a PIO: - caused by a timeout - caused by another reason	 602 601	 ERR BIT field gives bits on error

IA21 - POLL FUNCTION OF THE RDV (PHASE 2) TEST

This routine tests the continuity of the signal 'APR-I/APR-O' from one RDV to the next. The Poll command along with the level 1 remember and the level 1 latch are used to perform this checking.

This routine uses a table having 8 entries; each entry contains:

- the data to write
- the mask
- the expected data
- level 1
- the associated RAC (if error).

The RDVs diagnostic register allows the routines to perform all these functions.

ERC	Function	Error Description	RAC	Comments
7146	First Entry processing: Write in diag reg L1: - remember latch (1.6) - APR-I (1.0). Issue a Poll command. The RDVn should respond with its address.	Error occurred on the Poll.	602	ERR BIT field gives bits on error
7147	The diag reg should contain APR-O (1.5)	The diagnostic reg does not contain the expected data	602	ERR BIT field gives bits on error
7148	Second entry processing: Write into diag reg LVL1: - pending (1.7) and - APR-I (1.0) Issue a Poll command on which a timeout is expected	No timeout occurred on the Poll command	604	ERR BIT field gives bits on error
7149	The diag reg should have APR-S on	The diagnostic reg does not contain expected data	602	ERR BIT field gives bits on error
714A	3rd Entry processing Write into diagnostic reg: - APR-I (1.0) Issue a Poll command	A timeout is expected and it does not occur		
714B	The diag reg should have APR-O on	The diagnostic reg does not contain expected data	602	ERR BIT field gives bits on error
714C	Fourth entry processing: Write into diagnostic reg: - L1 remember (1.6) - L1 Pending (1.7) - APR-I (1.0) Issue a Poll command	Timeout occurred on the Poll or another RDV responded to Poll	602	ERR BIT field gives bits on error
714D	Diagnostic reg should be empty	The diag reg is not empty.	602	ERR BIT field gives bits on error

(IA21, continued)

ERC	Function	Error Description	RAC	Comments
714E	Fifth Entry processing: Write into diag register: - LVL1 remember (1.6) Issue a Poll command	Timeout is expected on Poll and does not occur	609	ERR BIT field gives bits on error
714F	Diagnostic reg should have APR-0 (1.5) on	APR-0 (1.5) is not on	602	ERR BIT field gives bits on error
7150	Sixth Entry processing: Write into diag register: - LVL1 pending (1.7) Issue Poll command.	Timeout as expected does not occur or another RDV responded to Poll	609	ERR BIT field gives bits on error
7151	Diagnostic reg is expected to be zero	Diagnostic reg is not empty	602	ERR BIT field gives bits on error
7152	Seventh Entry processing: Reset diag reg and issue a Poll with timeout expected command	Timeout as expected does not occur or another RDV responded to Poll	609	ERR BIT field gives bits on error
7153	Diagnostic reg is expected zero	Diagnostic reg is not empty	602	ERR BIT field gives bits on error
7154	Eighth Entry processing: Write into diag reg L1 remember latch (1.6) L1 pending (1.7) Issue a Poll and expect timeout	The timeout does not occur or another RDV responded to the Poll command.	609	ERR BIT field gives bits on error
7155	Diag reg should have APR-0 and APR-5 off	Diag reg does not have APR-5 and APR-0 off	602	ERR BIT field gives bits on error
7156	Level 1 interrupt processing	Level 1 interrupt occurred on a PIO: - caused by timeout - caused by another reason	602 601	
		Timeout occurred and it is expected only for RDV4	605	

IA22 - POLL FUNCTION OF THE RDV (PHASE 3) TEST

Using the Poll command, the routine tests the distribution of the 'APR' signal from one RDV to the next. The test is applied by RDV pairs in the following order: RDV1-RDV2, RDV2-RDV3, RDV3-RDV4, and so on up to RDV8-RDV9.

The L1 remember latch (1.6) of the diagnostic register is set on for each adjacent RDV of the pair; the RDV should respond to the Poll command with the address of the second redrive placed on the bus.

ERC	Function	Error Description	RAC	Comments
715B 715C	The following sequence of PIOs is performed for each pair of RDVs: - Diag Write RDV1 with all bits zeros - Diag Read RDV1 APR-O (1.5) on - Diag Write RDV2 L1 Remb. (1.6) - Poll gives RDV2 address - Reset diag reg	APR-O is not on for RDV1 The RDV address received by Poll is not the expected one APR-O is not off for RDVn	602 602 608	ERR BIT field gives bits on error ERR BIT field gives bits on error ERR BIT field gives bits on error
715E	Level 1 interrupt processing	Level 1 interrupt occurred on a PIO: - caused by a timeout - caused by another reason	 602 601	

IA23 - POLL FUNCTION OF THE RDV (PHASE 4) TEST

This routine tests the level 1 and Poll functions of the secondary bus (in fact, the signal 'APR-S' is only used by the secondary bus, therefore this routine never executes on the primary bus).

The routine writes the APR-I bit (1.0) into the diagnostic register to have APR-S propagated, then starts the checking with RDV5.

ERC	Function	Error Description	RAC	Comments
7163	Write into diag reg the L1 remember latch (1.6) then check APR-O (1.5) is off	In diag reg APR-O is found on	602	ERR BIT field gives bits on error
7164	Reset diag reg and check that APR-I (1.0) is on	In diag reg APR-I is found off	608	ERR BIT field gives bits on error
7165	Level 1 interrupt processing	Level 1 interrupt occurred on a PIO: - caused by a timeout - caused by another reason	 602 601	

IA30 - TEST THE READ ERROR REGISTER BROADCAST ON SECONDARY RDVS

This routine tests the capability of a secondary redrive to respond to a Read Error Register Broadcast command when the error register of that redrive is non-zero.

Note: the Read Error Register Broadcast command behaves as a Poll and as such the RDV having any bit on in its register responds by placing its address in byte zero and its error register in byte 1.

ERC	Function	Error Description	RAC	Comments
716A	PIOs on each secondary RDV: - Write Error Reg (RDVn) using data X'FF' - Read Diag reg (RDV4) - expected APR-5 (1.4) on - Read Error Reg Broadcast expected: Byte 0 RDVn addr Byte 1 X'FF' - Reset RDVn Error Reg - Read Error Reg Broadcast expected timeout - Reset RDV4 diag reg	RDVn error Note: RDVn, n is from 4 to 9 ----	602	ERR BIT field gives bits on error
716C	Level 1 interrupt processing when timeout is not expected	Level 1 interrupt occurred on a PIO: - caused by a timeout - caused by another reason	602 601	
716B	Level 1 interrupt processing when timeout is expected	Level 1 interrupt occurred and is not identified as a timeout	602 602	

IA40 - AUTOSELECTION (PHASE 1) TEST

This routine tests the autoselection mechanism for redrive 1 only. It uses a table having 8 entries. Each entry contains the data to be written, the expected data on the Read command, and the address of isolation sequence.

Pattern written	Pattern expected
X'00'	X'00'
X'38'	X'00'
X'40'	X'02'
X'58'	X'02'
X'60'	X'01'
X'68'	X'02'
X'78'	X'01'
X'00'	X'00'

The diagnostic register is used to test the autoselection function.

ERC	Function	Error Description	RAC	Comments
7170 to 7177	Take first/next pattern from the table, enable redrive 1 and write the pattern into the diagnostic register. Check the result when the Diagnostic Read is done. Note: Each ERC corresponds to a given pattern.	When error is found, get the address of isolation sequence - Error occurred on patts. 1, 2, 3, 4, or 5. - Error occurred on pattern 6 or 7. Retry is done with RDV disabled - If error then..... - Otherwise.....	 602 602 605	
718C	Level 1 interrupt processing	Level 1 interrupt occurred - caused by a timeout - caused by another reason	 602 601	

IA41 - AUTOSELECTION (PHASE 2) TEST

This routine tests the autoselection mechanism on all RDVs except RDV1. It uses a table having 12 entries. Each entry contains the data to be written, the expected data on the Read command, and the address of isolation sequence.

Pattern written	Pattern expected
X'00'	X'00'
X'38'	X'00'
X'40'	X'00'
X'78'	X'00'
X'04'	X'00'
X'3C'	X'00'
X'44'	X'02'
X'5C'	X'02'
X'64'	X'01'
X'6C'	X'02'
X'7C'	X'01'
X'00'	X'00'

The diagnostic register is used to test the autoselection function.

ERC	Function	Error Description	RAC	Comments
7180 to 718B	Take the next pattern from the table. Enable RDVn and write the pattern into the diagnostic register Check the result when Diagnostic Read is done. Note: Each ERC corresponds to a given pattern.	When error is found, get the address of isolation sequence - Error occurred on patts. 1 through 9 or 12. - Error occurred on patts. 10 and 11 Retry is done with RDV disabled - If error then..... - Else.....	 602 602 605	 ERR BIT field gives bits on error
718C	Level 1 interrupt processing	Level 1 interrupt occurred - caused by a timeout..... - caused by another reason	 602 601	

IA42 - AUTOSELECTION (PHASE 3) TEST

This routine tests the signals 'Select Out Preliminary' (SOP) and 'Select Out Secondary' (SOS). It uses the diagnostic register of the RDVs to perform the test. For all RDVs the bit TA Diag is set on (1.1), and read back via a PIO Read, the SOP bit (1.6) is expected to be on.

However there is an exception for RDV4. TA and L2 (1.1, 1.2) are written into the diagnostic register; the SOS bit (1.7) is expected to be on. For the second step only the secondary bus is checked for the SOS bit (1.7).

ERC	Function	Error Description	RAC	Comments
7191	For each RDV, write the TA bit (1.1) into diagnostic register and read it back. SOP (1.6) is expected to be on.	Error found when checking diagnostic reg via PIO Read	607	ERR BIT field gives bits on error
7190	For each RDV, write X'44' into the diag register. The expected result is: For RDV4: X'41' For RDVn: X'42'	Error found when checking diagnostic reg via PIO Read	602	ERR BIT field gives bits on error
7196	For each secondary RDV, write X'64' into diagnostic register. The expected result is: X'01'	Error found when checking diagnostic register	608	ERR BIT field gives bits on error
7192	Level 1 interrupt processing for 2 first items.	Level 1 interrupt occurred - caused by a timeout - caused by another reason	 602 601	
7197	Level 1 interrupt processing for secondary bus test	Level 1 interrupt occurred - caused by a timeout - caused by another reason	 602 601	

IA50 - TEST THE RESET COMMAND (SPECIFIC) TO RDVS

This routine tests the RESET command on each RDV. Both RDV registers (error register and diagnostic register) are set to X'FF'. Then a Reset Specific command is sent to each RDV; the result is verified:

- The error register of the addressed RDV should have zeros in byte 1 and the RDV address in byte 0.
- The diagnostic register of the addressed RDV should be zero.

ERC	Function	Error Description	RAC	Comments
719D	For each RDV do the following: - Write X'FF' into error register - Write X'FF' into diagnostic register - Send RESET command to RDVn - Read Error Register Expected result is: byte 0 = RDVn address byte 1 = 00	Error found when checking the result of error reg after RESET	602	ERR BIT field gives bits on error
719F	For each RDV read the diag reg and check the result (expected to be zero)	Error found when checking the result of diag reg after Reset	602	ERR BIT field gives bits on error
71A0	Level 1 interrupt processing	Level 1 interrupt occurred on a PIO: - caused by a timeout - caused by another reason	602 601	

IA51 - TEST THE RESET COMMAND (BROADCAST) TO RDVS

This routine tests the Reset command (broadcast) using the error register of the RDV and the following sequence of commands:

- Write Error Register Broadcast with data pattern X'FF'.
- Reset Broadcast
- Read Error Register Broadcast
- The last PIO Read should complete by timing out.

ERC	Function	Error Description	RAC	Comments
71A4	Do the following broadcast commands: - Write Error Register - Reset - Read Error Register A timeout should occur if all RDVs are reset	No timeout occurred after the PIO, Read Error Reg after Reset	602	
71A5	Level 1 interrupt processing	Level 1 interrupt occurred and is not expected: - caused by a timeout - caused by another reason	602 601	

IA60 - ENABLE/DISABLE COMMANDS (PHASE 1) TEST

This routine tests the Enable/Disable commands using the error register of the RDV. The Enable/Disable state of the redrive is provided by bit 0.1 of the register. The following sequence of PIO commands is performed for each RDV:

- Disable.
- Read Error Register and check that bit 0.1 is on.
- Enable.
- Read Error Register and check that bit 0.1 is off.
- Disable.
- Read Error Register and check that bit 0.1 is on.
- Enable.

ERC	Function	Error Description	RAC	Comments
71A8	For each RDV do the following: - Disable RDVn - Read Error Register Check that bit 0.1 is on	The disable bit is not found on	602	ERR BIT field gives bits on error
71A9	For each RDV do the following: - Enable RDVn - Read Error Register Check that bit 0.1 is off	The enable/disable bit is not as expected	602	ERR BIT field gives bits on error
71AA	For each RDV do the following: - Disable RDVn - Raed Error Register Check that bit 0.1 is on	The enable/disable bit is not found as expected	602	ERR BIT field gives bits on error
71AD	Level 1 interrupt processing	Level 1 interrupt occurred on a PIO: - caused by a timeout - caused by another reason	 602 601	

IA61 - ENABLE/DISABLE COMMANDS (PHASE 2) TEST

This routine tests the Enable/Disable commands for the secondary bus through RDV4. The routine disables RDV4 and issues a PIO command to every installed RDV on the secondary bus (RDV5 through RDV9). The PIO should complete by timeout because of the isolation of the secondary bus.

ERC	Function	Error Description	RAC	Comments
71B0	Disable RDV4 and issue a PIO to every RDV of the secondary bus (5 through 9). A timeout is expected when addressing every secondary RDV.	A timeout is expected for RDVn and does not occur	602	
71AD	Level 1 interrupt processing	Level 1 interrupt occurred on a PIO: - caused by an unexpected timeout - caused by another reason	 602 601	

IA62 - ENABLE/DISABLE COMMANDS (PHASE 3) TEST

This routine tests that the Reset RDV command has no effect on the Enable/Disable state of the RDV. The following sequence of PIO commands is sent for each RDV:

- Disable Broadcast.
- Reset Broadcast.
- Read Error Register of each RDV and check that bit 0.1 is on.
- Enable Broadcast.
- Reset Broadcast.
- Read Error Register of each RDV and check that bit 0.1 is off.

ERC	Function	Error Description	RAC	Comments
71B6	For each RDV do the following: - Disable Broadcast - Reset Broadcast - Read Error Reg Specific of RDVn Check that error reg of RDVn has bit 0.1 on	When checking the RDVn error reg, it is not found as expected.	602	ERR BIT field gives bits on error
71B7	For each RDV, do the following: - Enable Broadcast - Reset Broadcast - Read Error Reg Specific of RDVn Check that error reg of RDVn has bit 0.1 off	When checking the RDVn error reg, it is not found as expected.	602	ERR BIT field gives bits on error
71B8	Level 1 interrupt processing	Level 1 interrupt occurred on a PIO: - caused by a timeout - caused by another reason	 602 601	

IA70 - TEST THE CAPABILITY OF THE RDV TO DECODE COMMANDS

This routine tests the capability of each RDV to reject commands which are not included in the command set of the redrive.

The routine uses a table containing all invalid commands which are sent in turn to the addressed RDV.

The set of invalid commands placed on the bus at TA time is:

X'4021', X'4031', X'40X'41', X'4051', X'4061', X'4070', X'4071', X'4081', X'40B1', X'40C1', X'40D1', X'40E1', X'40F0', and X'40F1'.

If the RDV decodes an invalid command, it sets error register bit 1.7 on.

ERC	Function	Error Description	RAC	Comments
71BD to 71CA	Take first/next invalid cmd from the table and do the following: - Reset Redrive Specific to RDVn - Issue Invalid cmd to Rdvn - Check that error reg bit 1.7 is on for RDVn Note: Each invalid cmd is associated with an ERC in the range 71BD through 71CA	RDVn does not reject an invalid command bit 1.7 of error reg not found on	602	ERR BIT gives bits on error
71CB	Level 1 interrupt processing	Level 1 interrupt occurred on PIO: - caused by a timeout - caused by another reason	602 601	

IA71 - TEST READ ERROR REGISTER BROADCAST

This routine tests the capability of each RDV to respond to a Read Error Register Broadcast when its error register has a bit set on. Each entry contains the data to be written, and the expected data.

Pattern written (data bus byte 0)	Pattern expected (data bus byte 1)
X'80'	X'80'
X'40'	X'40'
X'20'	X'20'
X'10'	X'10'
X'08'	X'08'
X'04'	X'04'
X'02'	X'02'
X'01'	X'01'

The routine checks both the expected data pattern and the address of the RDV (contained in byte 0).

ERC	Function	Error Description	RAC	Comments
71D0 to 71D7	Take first/next data pattern from the table and do the following for each RDV: - Write Diag Reg APR-I (1.0) - Write data pattern into error reg of RDVn - Read Error Reg Broadcast Check byte 0 = RDVn address Check byte 1 = expected data pattern Note: Each data pattern is associated with an ERC in the range 71D0 through 71D7 - Reset redrive	Error in checking RDV address and data pattern	602	ERR BIT field gives bits on error
71D8	Level 1 interrupt processing	Level 1 interrupt occurred on PIO: - caused by a timeout - caused by another reason	 602 601	

IAIH - LEVEL 1 INTERRUPT HANDLER

The IFT IA handles only level 1 interrupts caused by the IOC all other interrupt levels are masked including level 1 interrupts raised by the adapters. The interrupt handler issues In X'7E' and In X'76' to determine the origin of the interrupt; if it is an IOC L1, then normal processing takes place. Otherwise, an ABEND is requested to terminate the request.

ERC	Function	Error Description	RAC	Comments
72FF	Process IOC level 1 when the IH L1 is dispatched and the IFT has not issued any PIO Note: No address field is provided, due to the direct entry into level 1.	L1 dispatched prior to any routine being started.	601	ERR bits is the image of In '76'
71FF	Interrupt level 1 is IOC but when resetting the origin of interrupt L1 is redispatched	Cannot reset IOC level 1 and therefore cannot exit to return to level 4.	646	
	Do non-IOC level 1 interrupt processing	Terminate the request by ABEND. L1 is not IOC.	670	

JA - LEVEL 1 INTERRUPT HANDLER

The level 1 interrupt handler is dispatched when the DCF identifies a level 1 interrupt and the IOC bus diagnostics are running. Two types of level 1 interrupt are processed:

1. IOC level 1.
2. Adapter level 1 (from the CSP).

In all other cases, an ABEND 670 is requested and terminates the section.

ERC	Function	Error Description	RAC	Comments
7F7F	Process IOC level 1 case 1	An IOC level 1 occurred and no I/O operation is pending	642	
77C0	Process IOC level 1 case 2	IOC level 1 occurred. Tried to reset it but level has been redispatched with no control returned to routine.	642	ADDIT INFO gives adapters
77C1	Process adapter level 1 case 1	An adapter level 1 occurred A Get Error Status has been sent to CSP to clear level 1 but it is still pending	642	ADDIT INFO gives adapters or MSG 'LABC'
77C2	Process adapter level 1 case 2	An adapter level 1 occurred A Get Error Status has been sent to both CSPs of the board to clear level 1 but it is still pending - If it is RDV3 then..... - If it is RDV7 then..... - For all other RDVs.....	 627 628 629	 ADDIT INFO gives adapters or MSG 'LABC'

JA - LEVEL 2 INTERRUPT HANDLER

The level 2 interrupt handler is dispatched when the DCF determines that a level 2 interrupt request to CCU has been raised and that level is not masked. The interrupt handler determines the origin of the level 2 interrupt via an In X'77' instruction. If it was an adapter level 2, it issues a 'Get Line ID' instruction to clear the interrupt. If it was not an adapter level 2 interrupt, it issues an ABEND 671 and terminates the section.

ERC	Function	Error Description	RAC	Comments
77C3	Process level 2 interrupt when no I/O is pending	A level 2 interrupt occurred and no I/O is in progress.	645	ADDIT INFO gives adapters or MSG 'LABC'
77C4	Process level 2 interrupt when it cannot be cleared by a 'Get Line ID' command sent to the CSP.	A level 2 interrupt occurred A 'Get Line ID' has been sent but interrupt is still pending. - If it is RDV3 then - If it is RDV7 then - In all other cases	 62C 62F 62E	 ADDIT INFO gives adapters or MSG 'LABC'

JA80 - IOC BUS TEST PHASE 1

This routine tests the following IOC bus functions:

- PIO Write to the CSP
- PIO Read from the CSP
- AIO (Indirect) data patterns: X'FFFF', X'0000', X'FFFF', X'B7DC', X'0269', X'B7DC', X'0269', and X'FFFF' from the CCU to the CSP.
- L1 and L2 interrupts from adapter

The routine communicates with the ROS microcode of the CSP adapter to perform the test. The ROS microcode performs the necessary tag checking (refer to the CSP ROS Diagnostic Description). The checking of the data pattern transmitted by the AIO is done by the ROS diagnostic microcode.

If an error is reported by JA80, and a TRM is in the second adapter, the message "LABC" is displayed in the ADDIT INFO field. The CE should run JC diagnostics to determine if the error is caused by the TRM, and use the IOC bus replacement procedure (see MIM-2) to fix the failure.

If an error is reported by JA80, and no message in the ADDIT INFO field, the CE should plug the CELIA card before attempting any repair procedure.

Note: If an error occurs in this routine, the remaining routines of the IOC bus IFT must not be run. The diagnostic abends if you try to do so. Press T to leave the routine.

ERC	Function	Error Description	RAC	Comments
7719	Perform Write PIO to CSPn with data pattern X'FFFF' (checked by ROS microcode) and level 2 interrupt is expected. If LABB then do the same PIO before analysis.	Level 2 interrupt expected and not received - If LABB and error occurred on both CSPs then.....	664	ADDIT INFO gives adapter nos.
		- If LABA then.....	61A	
		- If LABC then an error was detected and can possibly be caused by the TRM	61A	ADDIT INFO 'LABC'
7721	PIO Write to CLAB	- If CLAB then.....	61E	
7719	Perform Read PIO to CSPn and adapter level 1 is expected. If LABB board then do the same test for second CSP	Adapter level 1 interrupt expected and not received - If LABB and error occurred on both CSPs then.....	662	ADDIT INFO gives adapter nos.
		- If LABA then.....	616	
		- If LABC then an error was detected and can possibly be caused by the TRM.....	616	ADDIT INFO 'LABC'
7721	PIO Read on CLAB Perform AIO Indirect, transmit the set of patterns (8 halfwords) to the CSP(s) and expect an L2 interrupt request from the adapter	- if CLAB then.....	617	

(JA80, continued)

ERC	Function	Error Description	RAC	Comments
7711	Check the ending conditions for the AIO	Level 1 interrupt received from adapter:		
		- If it is a LABB then.....	664	ADDIT INFO gives
		- If it is a LABA then.....	61A	adapter nos.
		- If it is a CLAB then.....	61E	
		- If it is a LABC then.....	61A	ADDIT INFO 'LABC'
7711	Check the received level 1 after AIO	IOC level 1 received (timeout):		
		- If it is a LABB then.....	662	ADDIT INFO gives
		- If it is a LABA then.....	616	adapter nos.
		- If it is a CLAB then.....	617	
		- If it is a LABC then.....	61A	ADDIT INFO 'LABC'
7711	Check the ending conditions for the AIO	Level 1 interrupt received from adapter:		
		- If it is a LABB then.....	664	ADDIT INFO gives
		- If it is a LABA then.....	61A	adapter nos.
		- If it is a CLAB then.....	61E	
		- If it is a LABC then.....	61A	ADDIT INFO 'LABC'
7720	Check cycle steal pointer register when AIO complete	Transmission of data is stopped:		
		- If it is a LABB then.....	660	ADDIT INFO gives
		- If it is a LABA then.....	618	adapter nos.
		- If it is a CLAB then.....	619	
		- If it is a LABC then.....	61A	ADDIT INFO 'LABC'

JA81 - IOC BUS TEST PHASE 2 AND CSP RESPONDER LOADING

This routine tests the AIO (cycle steal to CSP adapter) using the CSCW Direct/Indirect Long. The routine sends a responder microcode to each CSP to perform all other tests required by the IOC bus IFT. The CSP acknowledges the loading by sending a status of X'8000'.

To perform the loading, the routine communicates with the ROS diagnostic microcode.

If an error is reported by JA81, the CE should plug the CELIA card before attempting any repair procedure.

Note: If an error occurs in this routine, the remaining routines of the IOC bus IFT must not be run. The diagnostic abends if you try to do so. Press T to leave the routine.

ERC	Function	Error Description	RAC	Comments
7722	Request the CSP to initiate the AIO Direct/Indirect, Long and wait for level 2 interrupt to come.	After timer elapsed no level 2 interrupt received: - If it is a LABB then..... - If it is a LABA then..... - If it is a CLAB then.....	 660 618 619	 ADDIT INFO gives adapter nos.
7723	Check the ending status sent by the CSP. Expected status is X'8000'	Status is wrong meaning error during the load: - If it is a LABB then..... - If it is a LABA then..... - If it is a CLAB then.....	 665 616 617	 ADDIT INFO gives adapter nos.
	Level 1 interrupt analysis:	Report level 1 error		
7724 7725	Adapter level 1 received IOC level 1 received	- If it is a LABB then..... - If it is a LABA then..... - If it is a CLAB then.....	665 616 617	ADDIT INFO gives adapter nos.

JA82 - TEST IOH TO CSP WITH R2 = 0

This routine tests the capability of the IOC to address an adapter when R2 is zero. The IOC then uses local store X'48' to send the TA value on the bus. The mechanism used by the routine is as follows:

- Send an IOH with R2 = 0 and an invalid TA in LS X'48'. A timeout and an L1 interrupt are expected.
- Send an IOH Write with R2 = 0 and valid TA and TD (X'5555').
- Send an IOH Read with R2 = 0. The data pattern expected is X'5555'.

ERC	Function	Error Description	RAC	Comments
7730	Set up LS X'48' with an invalid TA (Write), then issue IOH with R2 = 0 and expect an IOC level 1 time-out	No level 1 occurred with an invalid TA value	61D	ADDIT INFO gives adapters
7735	Set up LS X'48' with an invalid TA (Read), then issue IOH with R2 = 0 and expect an IOC level 1 timeout	No level 1 occurred with an invalid TA value	613	ADDIT INFO gives adapters
7731	Set up LS X'48' with a valid TA value, then issue an IOH with R2 = 0 and with data pattern X'5555'. Read back the sent data and check that it is X'5555'.	Mismatch in the data pattern	61A	ADDIT INFO gives adapters
7733	The function of this ERC is associated with the RAC sent by the responder. Refer to the responder description.			ADDIT INFO gives adapters

JA83 - TEST IOHI TO CSP

The immediate value of the IOHI instruction is the TA value and the Read or Write bit. The routine uses a table of subroutines to test this function.

The table contains 2 entries per CSP, one for Write and one for Read; the table is built in such way that all possible CSP adapter address are implemented. The data written is X'5555'. The routine then checks that the Read returns the same data patterns.

ERC	Function	Error Description	RAC	Comments
7740	Issue IOHI Write with data pattern X'5555' and read it back via an IOHI Read command. Check that received matches.	Received data pattern is not the same as the one sent	61A	ADDIT INFO gives adapters
7742	Process level 1 interrupt received	A level 1 interrupt occurred - If it is an IOC level 1 then.....	616	ADDIT INFO gives adapters
		- If it is an adapter L1 then see the responder description		
7724	Adapter level 1 received	- If it is a LABB then.....	665	ADDIT INFO gives adapter nos.
7725	IOC level 1 received	- If it is a LABA then.....	616	
		- If it is a CLAB then.....	617	

JA90 - TEST AIO DIRECT READ

To test the AIO, the routine sends the required parameters to the adapter via a PIO. The TD value tells the responder which AIO must be initiated.

The routine then requests an AIO Read, Short, Direct for one halfword in the first step, followed by an AIO Read, Long, Direct for one halfword. In both cases, the routine expects a level 2 interrupt and then checks the CPR (channel pointer register).

ERC	Function	Error Description	RAC	Comments
7745	Request the CSPn to initialize AIO Read, Short, Direct and expect a level 2 interrupt from CSPn	No level 2 interrupt from the CSP	618	ADDIT INFO gives adapters
7747	Check NCPR value (basic channel pointer register from In X'3F')	Incorrect BCPR from the CSP	61F	ADDIT INFO gives adapters
7748	Request the CSPn to initialize AIO Read, Long, Direct and expect a level 2 interrupt from CSPn	No level 2 interrupt from the CSP	618	ADDIT INFO gives adapters
7749	Check BCPR and ECPR values from In X'3F'	Incorrect BCPR and ECPR from the CSP	61F	ADDIT INFO gives adapters
774B	Process level 1 interrupt	A level 1 interrupt occurred - If it is an IOC level 1 then..... - If it is an adapter level 1 then see the responder description	616	ADDIT INFO gives adapters

JA91 - TEST AIO DIRECT WRITE

To test the AIO, the routine sends the required parameters to the adapter via a PIO. The TD value describes to the responder which AIO must be initiated. The routine requests an AIO Write, Short, Direct/Indirect for one halfword (X'A5A5'), followed by a request an AIO Write, Short, Direct/Indirect for one halfword (X'A5A5'), and then a request for an AIO Read, Short, Direct/Indirect. After each AIO a level 2 interrupt is expected. When the AIO Read is complete, the data is checked.

ERC	Function	Error Description	RAC	Comments
7750	Request the CSPn to initialize AIO Write, Short Direct/Indirect with data X'A5A5' and expect a level 2 interrupt.	No level 2 interrupt from the CSP	618	ADDIT INFO gives adapters
7752	Request the CSPn to initialize AIO Read, Short, Direct/Indirect and 2 bytes. Expect a level 2 interrupt.	No level 2 interrupt from the CSP	618	ADDIT INFO gives adapters
7754	Compare the data sent and received	Mismatch in data compare	61A	ADDIT INFO gives adapters
7754	Process level 1 interrupt	A level 1 interrupt occurred - If it is an IOC level 1 then..... - If it is an adapter level 1 then see responder description	616	ADDIT INFO gives adapters

JA92 - TEST THAT INVALID CHCW CAUSES A LEVEL 1 INTERRUPT

This routine requests the CSP to initialize an AIO with an invalid CHCW (channel control word). When the IOC recognizes the invalid CHCW, it raises an IOC level 1 interrupt, latches the error in In X'76', and sends a halt tag to the adapter. The CSP then requests a level 1 interrupt and sends a status of X'4000' to acknowledge the halt.

ERC	Function	Error Description	RAC	Comments
7760	Request the CSPn to initialize an AIO with an invalid CHCW and expect a level 1	Incorrect CHCW does not cause a level 1 interrupt	613	ADDIT INFO gives adapters
7762	Check the status sent by the CSP to the CCU and expect X'4000'	Bad status received from the CSP	61A	ADDIT INFO gives adapters

JA93 - TEST ADDRESS EXCEPTION USING AIO

This routine requests the CSP to initialize an AIO with a CCU address defined as not implemented. The routine sets up the address exception via an Out X'73' with value X'1E069' (Enable Address Exception Key for address X'1E000'). The address is sent to the CSP via PIO commands.

The CSP requests an AIO Write, Long, Direct/Indirect and 4 bytes of data; a level 1 interrupt is then expected from the IOC and from the adapter. After all checking has been done, restore the storage installed.

ERC	Function	Error Description	RAC	Comments
7766	Request CSPn to initialize an AIO with a data transfer into a block of storage having the address exception key set on and check for a level 1 interrupt.	No level when CSP transmit data into storage having Address Exception (not installed)	613	ADDIT INFO gives adapters
776A	Check the status sent by the responder (halt received) status: X'4000'	Bad status received from the CSP	61A	ADDIT INFO gives adapters

JA94 - TEST STORAGE PROTECT VIOLATION BY ADAPTER USING AIO

This routine requests the CSP to initialize an AIO to transmit data into a block of protected CCU storage (not having the same protect key). It enables the storage protect key for the block starting at address X'C000' by an Out X'73' with X'0C079'.

A level 1 interrupt is expected for the storage protect violation; the CSP sends the status X'4000' to acknowledge the halt sent by the IOC to the adapter.

ERC	Function	Error Description	RAC	Comments
7770	Request CSPn to initialize an AIO and to transmit data into storage location X'C00' which has been set to the protected condition. Expect a level 1 interrupt	No level 1 when an adapter transmit data into storage protected	613	ADDIT INFO gives adapters
7772	Check the status sent by the responder (halt received) Status: X'4000'	Bad status received from CSP	618	ADDIT INFO gives adapters

JA95 - TEST AIO WITH DATA ON BYTE BOUNDARY

This routine tests the capability for an AIO to transfer data with an odd CCU storage address. The routine sends an odd CCU storage address to the adapter as BCPR and requests the CSP to initialize an AIO Write, Short, Direct/Indirect with a count of 4 bytes (X'A5A5A5A5'). A level 2 interrupt is expected. The routine requests the responder to send back via an AIO Read, Short, Direct/Indirect, the same 4 bytes. A level 2 interrupt is expected. The routine then checks the sent and received data.

The same run is then done with an even CCU storage address; the same checking is done.

ERC	Function	Error Description	RAC	Comments
7776	Send via PIO an odd CCU storage address as BCPR to the CSP and request the CSP to initialize an AIO Write, Short, D/I for 4 bytes and expect a level 2 interrupt. Data: X'A5A5A5A5'	No level 2 received from CSP	618	ADDIT INFO gives adapters
7777	Request the CSP to initialize an AIO Read, Short, D/I for 4 bytes sent by previous AIO. Expect a level 2 interrupt and the 4-byte pattern X'A5A5A5A5'.	No level 2 received from CSP	618	ADDIT INFO gives adapters
777B	Check that sent and received data compare with no error	Mismatch in the data	613	ADDIT INFO gives adapters
7779	Request the CSP to initialize an AIO Write, Short, D/I for 4 bytes X'A5A5A5A5' even BCPR and expect a level 2 interrupt from the CSP.	No level 2 received from CSP	618	ADDIT INFO gives adapters
777D	Request the CSP to initialize an AIO Read, Short, D/I for the 4 bytes sent by the previous AIO and expect a level 2 interrupt from the CSP.	No level 2 received from CSP	618	ADDIT INFO gives adapters
777E	Check that sent and received data compare with no error	Mismatch in the data	61A	ADDIT INFO gives adapters
7781 7783	Process level 1 interrupt	A level 1 interrupt occurred during an AIO - If it is an IOC level 1 then..... - If it is an adapter level 1 then see responder diagnostic description	616	ADDIT INFO gives adapters

JA96 - TEST AIO WITH CHCW HAVING A 'MOSS FLAG'

This routine tests the AIO with a CHCW having the bit 'MOSS originated'. It requests the CSP to initialize an AIO Write, Short, D/I, MOSS and 2 bytes of transmitted data, followed by an AIO Read, Short, D/I, MOSS and 2 bytes of data. The routine checks that a level 2 interrupt is received in the CCU after each AIO completes and verifies that the data match.

ERC	Function	Error Description	RAC	Comments
7777	Request the CSP to initialize an AIO Write, Short, D/I, MOSS for 2 bytes of data and check that a level 2 interrupt has been received by CCU.	No level 2 interrupt received by CCU	618	ADDIT INFO gives adapters
7779	Request the CSP to initialize an AIO Read, Short, D/I, MOS for 2 bytes of data and check that a level 2 interrupt has been received by the CCU.	No level 2 interrupt received by CCU	618	ADDIT INFO gives adapters
7786	Check that sent and received data match.	Mismatch in data compare	61A	ADDIT INFO gives adapters

JA98 - TEST AIO LONG

This routine requests the CSP to initialize an AIO Write, Long, D/I for the transfer of 2 bytes (X'A5A5'), waits for completion, and then requests the CSP to initialize an AIO Read, Long, D/I, for the transfer of the same 2 bytes and wait for completion.

For both AIOs the routine expects a level 2 interrupt request from the adapter and checks the sent and received data.

ERC	Function	Error Description	RAC	Comments
778C	Request the CSP to initialize an AIO Write, Long, D/I, for the transfer of A5A5 data pattern from CCU to CSP and wait for level 2 interrupt.	No level 2 interrupt received by the CSP	618	ADDIT INFO gives adapters
7787	Request to CSP to initialize an AIO Read, Long, D/I for the transfer of the previously sent data and wait for level 2 interrupt.	No level 2 interrupt received by the CSP	618	ADDIT INFO gives adapters
778A	Check that sent and received from CSP by second AIO	Mismatch in data compare	61A	ADDIT INFO gives adapters
7789	Process level 1 interrupt	Level 1 interrupt occurred during an AIO - If it is an IOC level 1 then..... - If it is an adapter level 1 then refer to diag description for the CSP responder	616	ADDIT INFO gives adapters

JA99 - TEST 'VALID BYTE AND MODIFIER' FROM THE CSP

The 'Valid byte and Modifier' is a combination of tags sent to the CCU-IOC at the end of AIO sequence instead of EOC. It informs the CCU that an odd number of bytes were transmitted. The routine requests the CSP to initialize AIO Write, Short, D/I for 3 bytes of data (X'AAAAAA'), followed by an AIO Read, Short, D/I, for 1 byte (X'AA') and wait for completion. It checks for a level 2 interrupt after each AIO and also checks that an odd number of bytes have been transmitted.

Note: If an error occurs in this routine, the remaining routines of the IOC bus IFT must not be run. The diagnostic abends if you try to do so. Press T to leave the routine.

ERC	Function	Error Description	RAC	Comments
7790	Request the CSP to initialize an AIO Read, Short, D/I for 3 bytes of data and wait for a level 2 interrupt from the CSP	No level 2 interrupt received by CCU	618	ADDIT INFO gives adapters
7786	Check that the data received is X'AAAAAA00'	Mismatch in data compare	61A	ADDIT INFO gives adapters
7795	Request the CSP to initialize an AIO Write, Short, D/I for 1 byte of data and wait for a level 2 interrupt from the CSP	No level 2 interrupt received from the CSP	618	ADDIT INFO gives adapters
779C	Process level 1 interrupt	Level 1 interrupt occurred during an AIO - If it is an IOC level 1 then..... - If it is an adapter level 1 then see diag description of the CSP responder	616	ADDIT INFO gives adapters

JA9A - TEST CSP 'HARD STOP'

The CSP Hard Stop is set when a CSP level 0 interrupt request occurs when in level 0. This causes an adapter level 1 to be sent to the CCU. The routine requests the CSP responder to force a hard stop. It sent a PIO with a hard stop bit. The CSP then executes the function and the CCU expects an adapter level 1 interrupt. The routine clears the interrupt and sends PIO 'Program Reset' in order to restart the responder.

ERC	Function	Error Description	RAC	Comments
77A0	Send PIO with TD having Hard Stop request to the CSP and expect a level 1 interrupt	No level 1 from adapter on CSP Hard Stop	626	ADDIT INFO gives adapters
77A2	Check the status sent by the CSP on a Get status command Expected: X'0020'	Bad status received from CSP after a hard stop	626	ADDIT INFO gives adapters
77A4	Send a PIO Program Reset to the CSP in order to restart the responder and expect a level 2 interrupt	No level 2 interrupt received from the CSP	626	ADDIT INFO gives adapters
77A8	Process abnormal level 1 interrupt	An interrupt level 1 occurred on PIO Program Reset - If it is an IOC level 1 then..... - If it is an adapter level 1 then see diag description for the CSP responder	616	ADDIT INFO gives adapters

JAA0 - TEST BSC CHARACTER DECODE

The CSP logic decodes the following BSC characters: ETX, ETB, and ENQ, both in EBCDIC (X'03', X'26', X'2D'), and in ASCII (X'03', X'17', X'05'). It informs the code by setting a bit in CSP-XR00. The routine requests the CSP to initialize an AIO Write, Short, D/I to transmit a string of characters in which a BSC character is inserted. For each BSC character in the string, the CSP performs an AIO. After each AIO, the routine expects a level 2 interrupt from the CSP. If an error is found by the responder, this one sends a level 1 interrupt to the CCU.

ERC	Function	Error Description	RAC	Comments
77A7	Request the CSP to init. AIO Write, Short, D/I for the transfer of data pattern in which there is a BSC character and expect a level 2 interrupt. Each string is as follows: X'FFFF XX00' X'FFFF 00XX' Where XX is the BSC character	No level 2 interrupt from the CSP	618	ADDIT INFO gives adapters
77A8	Request the CSP to init. AIO Write, Short, D/I for the transfer of data pattern in which there is a BSC character and expect a level 2 interrupt. Each string is as follows: X'FFFF FFFF' X'FFFF FFFF' X'XX00 00XX' Where XX is the BSC character	No level 2 interrupt from the CSP	618	ADDIT INFO gives adapters
77AC	Process level 1 interrupt	A level 1 interrupt occurred on an AIO - If it is an IOC level 1 then..... - If it is an adapter level 1 then refer to diag description of CSP responder.	616	ADDIT INFO gives adapters

JAA1 - TEST IOC FUNCTION OF IOH QUEUING

The IOC logic is able to queue a PIO when it is busy.

The routine first requests the CSP to initialize an AIO Write, Short, D/I for the transmission of 16 bytes (transmitted data = X'0001' through X'0008'), followed by an AIO Read, Short, D/I for the reception of the same 16 bytes. A level 2 interrupt is expected after the completion of each AIO; the data is checked.

The routine then starts sending PIO Writes and Reads with no wait and checks that the data sent via a PIO Write matches the data received via a PIO Read.

The routine then clears the responder read area via an AIO Write, short, D/I, with 16 bytes of all zeros.

ERC	Function	Error Description	RAC	Comments
77B0	Request CSP to initialize AIO Write, Short, D/I, 16 bytes of data (X'0001' to X'0008') and expect a level 2 interrupt from the CSP	No level 2 interrupt from the CSP	618	ADDIT INFO gives adapters
77B2	Request CSP to initialize AIO Read, Short, D/I, read same 16 bytes of data sent previously and expect level 2 interrupt	No level 2 interrupt from the CSP	618	ADDIT INFO gives adapters
7784	Check sent and received data done via cycle steal and via PIO queuing mechanism	Mismatch in data pattern	61A	
77B3	Request CSP to initialize AIO Write, Short, D/I, with 16 bytes of all zeros. Expect a level 2 interrupt	No level 2 interrupt from the CSP	618	ADDIT INFO gives adapters
77B8	Process level 1 interrupt	Level 1 interrupt occurred		
		- If it is an IOC level 1 then.....	616	ADDIT INFO gives adapters
		- If it is an adapter level 1 then refer to the diagnostic description of the CSP responder.		

JARP - CSP-IOC BUS RESPONDER

The CSP-IOC bus responder is loaded into each CSP when the section JA is executed. Routine JA81 of this section loads the responder microcode into the CSP in order to test the remaining parts of the IOC bus.

Routines JA82 through JAA1 communicate with this responder by exchanging commands via PIO. The general format of a command sent by the above routines is:

Bit	Meaning
0	AIO short
1	Moss originated cycle steal
2	Halt expected (force error)
3	Transmit BSC ending characters
4	Reserved
5-7	Cycle steal command 0 through 7 (see format of CHCW)
8	AIO requested is a write
9	AIO requested is a read
10	AIO requested is long
11	AIO requested is direct
12	AIO requested is D/I
13	AIO requested is indirect
14-15	Cycle steal count

When a CCU level 1 interrupt is requested by the responder, it sends the following status to the CCU when the Command 'Get Error Status' is sent by JA level 1 JIH.

CSP status sent to IOC bus IFT

The CSP status is one halfword long, and has the following format:

Bit	Meaning
0	Loading OK
1	Halt expected tag received
2	Invalid command
3	Unexpected halt
4	IOC bus check
5	Preselect check
6	Unexpected interrupt in CSP
7	BSC character bad decode
8-15	RAC

The following RACs are sent by the CSP responder: X'20', X'21', X'22', X'23', X'40', X'45', and X'50'.

The CCU code when issuing REDH to report the error appends X'6' as the first digit, giving the following list of RACs: X'620', X'621', X'622', X'623', X'640', X'645', and X'650'.

The error found and associated with each RAC is as follows:

RAC	Meaning
620	Invalid command
621	Unexpected halt
622	IOC bus check
623	Preselection check
640	Unexpected CSP level 1 and level zero
645	BSC character decode fails
650	Error found while testing 'VB'

CHCW Format

0 Short, Read, Direct

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0

1 Short, Write, Direct/Indirect

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

2 Short, Read, Direct/Indirect

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0

3 Short, Write, Direct

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0

4 Long, Read, Direct

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0

5 Long, Read, Direct/Indirect

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0

6 Long, Write, Direct

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0

7 Long, Write, Direct/Indirect

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

The CHCW (channel control word) is built up by the CSP microcode to perform an AIO to transfer data to or from the CCU in cycle steal mode on the IOC bus.

There are two modes of communication between the CCU and the CSP adapters:

1. PIO mode
2. AIO mode

The CSP has 2 buffers connected to the IOC bus for data communication with the CCU. These 2 buffers are called ping and pong buffers. The microcode residing in the CSP fills or empties these buffers (depending on the direction of the data

transfer) allowing synchronization of the flow between the CCU and the CSP.

1. PIO mode

   In PIO mode the TA value is put in the ping buffer and the TD value in the pong buffer.

   For a Write PIO the microcode empties the ping and pong buffer into the CSP control store.

   For a Read PIO the microcode places the two bytes of data in the pong buffer.

2. AIO mode

   All AIO operations are initiated by the CSP. The microcode places the CHCW in the ping buffer, and the CCU memory address in the pong buffer. If the operation is AIO long, the second part of the address is placed in the ping buffer. The cycle steal is started when the CSP microcode raises CSR (cycle steal request tag) and the IOC returns CSG and raises the EOC (End of Chain) tag.

Adapter Level 1 Processing

When the CSP responder requests a CCU level 1 interrupt, the JA section issues a 'Get Error Status' PIO command.

The CSP then sends the status which has been built (see description above) to the CCU. Each JA routine analyzes the interrupt and status and reports the error.

The table below shows the relationship between the ERCs generated by the JA routines and the RACs originated by the responder.

Routine	ERC	RACs	Comments
JA82	7733	620-620-621-622-623-640	ERR bits give CSP status
JA83	7744	620-620-621-622-623-640	ERR bits give CSP status
JA90	774A	620-620-621-622-623-640	ERR bits give CSP status
JA91	7756	620-620-621-622-623-640	ERR bits give CSP status
JA95	7780 7782	620-620-621-622-623-640	ERR bits give CSP status
JA98	7788	620-620-621-622-623-640	ERR bits give CSP status
JA99	779A	620-620-621-622-623-640 650	ERR bits give CSP status VB test fails
JA9A	77A6	620-620-621-622-623-640	ERR bits give CSP status
JAA0	77AA	620-620-621-622-623-640 645	ERR bits give CSP status BSC char. test fails
JAA1	77B6	620-620-621-622-623-640	ERR bits give CSP status

JBB0 - TEST IOC BUS INTERCONNECTION WITH CHANNEL ADAPTER

This routine selects the first/next CA given in the CDF and issues PIOs whose ID value is obtained from a table: X'FFFF', X'0000', X'B7DC', X'0269', and X'0000'.

If there is more than one CA on the board, the routine correlates the test from one CA to the next to determine the type of error, if any.

ERC	Function	Error Description	RAC	Comments
77F1	Select CA X'n' and verify from In X'6E' that the CA has been properly selected	The CA did not select when issuing Out X'67'	624	ADDIT INFO gives adapters
77F7	Issue PIO (Out X'64', X'65') to the selected CA and check the data (In X'64', X'65')	Mismatch in the data	624	ADDIT INFO gives adapters
77FB	Board has more than one CA Issue PIO to each CA	Mismatch in the data for multiple CAs attached to RDV 10	625	ADDIT INFO gives adapters
77F4	Process level 1 for selection	Tried to select a CA and level 1 interrupt occurred	624	ADDIT INFO gives adapters
77F9	Process level 1 for PIO Write (Out X'64', X'65')	Level 1 interrupt occurred when PIO Write issued	624	ADDIT INFO gives adapters
77FA	Process level 1 for PIO Read	Level 1 interrupt occurred when PIO Read was issued	624	ADDIT INFO gives adapters
3168	Board has more than one CA	Mismatch in the data for multiple CAs attached to an RDV other than RDV 10.	648	ADDIT INFO gives adapters

JB - LEVEL 1 INTERRUPT HANDLER

The level 1 interrupt handler is dispatched by the CDF when a Level 1 interrupt occurs and section JB is running. The interrupt handler determines the origin of the interrupt (IOC or adapter) from the In X'7E'. If the interrupt is an IOC level 1, it is reset via an Out X'76'; if it is an adapter level 1, then a Reset CA is done, followed by an exit.

ERC	Function	Error Description	RAC	Comments
77C4	Process level 1 interrupt	Tried to reset level 1 interrupt but it is still pending	647	ADDIT INFO gives adapters

JC - LEVEL 1 INTERRUPT HANDLER

The level 1 interrupt handler is dispatched when the DCF identifies a level 1 interrupt and the IOC bus diagnostics are running. Two types of level 1 interrupt are processed:

1. IOC level 1.
2. Adapter level 1 (from the TRM).

In all other cases, an ABEND 670 is requested and terminates the section.

ERC	Function	Error Description	RAC	Comments
7F7F	Process IOC level 1 case 1	An IOC level 1 occurred and no I/O operation is pending	652	
77C0	Process IOC level 1 case 2	IOC level 1 occurred. Tried to reset it but level has been redispatched with no control returned to routine.	652	ADDIT INFO gives adapters
77C1	Process adapter level 1 case 1	An adapter level 1 occurred A Get Error Status has been sent to TRM to clear level 1 but it is still pending	652	ADDIT INFO gives adapters
77C2	Process adapter level 1 case 2	An adapter level 1 occurred A Get Error Status has been sent to the TRM of the board to clear level 1 but it is still pending - If it is RDV3 then..... - If it is RDV7 then..... - For all other RDVs.....	 64B 64C 64D	 ADDIT INFO gives adapters

JC - LEVEL 2 INTERRUPT HANDLER

The level 2 interrupt handler is dispatched when the DCF determines that a level 2 interrupt request to CCU has been raised and that level is not masked. The interrupt handler determines the origin of the level 2 interrupt via an In X'77' instruction. If it was an adapter level 2, it issues a 'Get Line ID' instruction to clear the interrupt. If it was not an adapter level 2 interrupt, it issues an ABEND 671 and terminates the section.

ERC	Function	Error Description	RAC	Comments
77C3	Process level 2 interrupt when no I/O is pending	A level 2 interrupt occurred and no I/O is in progress.	653	ADDIT INFO gives adapters
77C4	Process level 2 interrupt when it cannot be cleared by a 'Get Line ID' command sent to the TRA.	A level 2 interrupt occurred A 'Get Line ID' has been sent but interrupt is still pending. - If it is RDV3 then..... - If it is RDV7 then..... - In all other cases.....	 64E 64F 651	 ADDIT INFO gives adapters

JC01 - IOC BUS TEST TO TRA PHASE 1

This routine tests the following IOC Bus functions:

- IOH WRITE to the TRM DIAGNOSTIC Register
- IOH READ to the TRM DIAGNOSTIC Register
- Compare data patterns: X'0000', X'FFFB', X'CCC0', X'AAAA', X'5555' X'3333', X'B6D8, X'DB69', X'4920', X'2492', X'9249', X'0004', X'001A', X'001E', and X'0000'.
- Test for Level 1 and Level 2 interrupts

This routine checks the IOC bus interface up to the TRM DIAGNOSTIC register with selected patterns using the WRITE and READ DIAGNOSTIC register commands.

The register is tested with a series of patterns that have a '0' in the START bit position.

This prevents the diagnostic logic from forcing any errors that are specified in the remaining bits of the register.

Then the START bit position is tested by setting the other bits to '0', so that no error is specified and turning the START bit on and off.

ERC	Function	Error Description	RAC	Comments
77F2	Process level 1 from IOC during PIO write	Level 1 interrupt occurred when PIO write was issued	656	
77F3	Process level 1 from IOC during PIO read	Level 1 interrupt occurred when PIO read was issued	656	
77F6	Process level 2 for PIO write	Level 2 interrupt occurred when PIO write was issued	656	
77F7	Issue PIO to a TRM	Mismatch in the data	656	
77F8	Process level 2 for PIO read	Level 2 interrupt occurred when PIO read was issued	656	
77F9	Process ADAPTER Level 1 for PIO write	Level 1 interrupt occurred when PIO write was issued	656	
77FA	Process ADAPTER Level 1 for PIO read	Level 1 interrupt occurred when PIO read was issued	656	

JC02 - IOC BUS TEST TO TRA PHASE 2

This routine tests the following IOC Bus and TRA hardware functions:

- Forcing of IDLE state error in the diagnostic register
- IDLE STATE parity checker
- Level 2 interrupt checked
- Reset Level 2 interrupt after a GET LINE ID command
- GET LINE ID data saved
- Reset the Diag REG of the TRM and test the GET LINE ID data.
- Reset L2 error status in the TRM.

This code tests the ability of the DIAGNOSTIC REGISTER to force an error on the IDLE STATE.

The detection of an error (by the IDLE STATE parity checker) and reporting are also checked.

The DIAGNOSTIC REGISTER is set up to force a parity error on the internal bus byte 0.

A level 2 interrupt should be received from the TRM.

A GET LINE ID command is issued to reset the interrupt by the L2 IH.

GET LINE ID data tested.

The TRM is issued a Diag reg reset.

L2 ERROR status reg is reset.

ERC	Function	Error Description	RAC	Comments
77F4	Process unexpected Level 1 interrupt	Unexpected level 1 interrupt on write to diag reg to set up force Level 2 error	659	
77F5	Process unexpected Level 1 interrupt	Unexpected level 1 interrupt occurred on PIO write to reset the diag reg.	659	
77FB	Testing of GET LINE ID data	Mismatch of GET LINE ID data	659	
77FC	Diagnostic reg set up to force a Level 2 interrupt	No Level 2 interrupt generated	659	
77FD	Process level 2 interrupt	A Level 2 interrupt occurred on write to TRA diag reg to set up to for L2 error.	659	
77FE	Process unexpected Level 2 interrupt	A Level 2 interrupt occurred on the reset to the diag reg	659	
77FF	Process unexpected Level 2 interrupt on write to the buffer	A Level 2 interrupt occurred when PIO WRITE to TRA buffer command was issued.	659	

MB01 - SCOPING ROUTINE FOR THE IOC BUS

This routine provides the following scoping possibilities:

- PIO (IOH) to redrives, channel adapters, CSPs, and TRAs
- Channel adapter autoselection
- Cycle steal for channel adapters

1. PIO (IOH) to redrives, channel adapters, CSPs, and TRAs

   When the following message is displayed on the screen 'ENTER OPTION TA AND TD THEN PRESS SEND',

   The CE should enter the parameters in the following format:

   'R00aaaadddd' where

   00 stands for the option (01, 02 or 03)
   aaaa stands for the TA value
   dddd stands for the TD value

   Option 01 Executes the PIO once only, with error reporting, then returns to the prompt.

   Option 02 Loops on the PIO until an error is found and reported.

   Option 03 Loops on the PIO without error reporting.

   TA is 4 hexadecimal digits of data used to address an adapter and give the direction of the PIO (Read or Write).

   TD is 4 hexadecimal digits of data to write or to read to/from an adapter.

   Error reporting is described at the end of the routine description.

2. Channel adapter autoselection

   This routine tests the channel adapter autoselection mechanism. The PRI (program request interrupt) is used to perform this test.

   The test is done on a pair of channel adapters selected by the CE when the following prompting message is received:

   'TO TEST CA AUTOSELECT TYPE: CA THEN PRESS SEND'

   followed by the message:

   'ENTER CA NBRS FOR AUTOSELECT THEN PRESS SEND'

   The CE should enter the parameters in the following format:

   'R00xxyy' where

   00 stands for the option (01, 02 or 03)
   xx stands for the first CA number under test
   yy stands for the second CA number under test

   Option 01 Executes the test request once only, with error reporting

   Option 02 Loops on the test request until an error is found and reported.

   Option 03 Loops on the test request without error reporting.

   CA NBRS This is the pair of CAs to test. The range is from 01 to 06 and give the direction of the PIO (Read or Write).

If yy is equal to 00, only the first channel is tested.

Error reporting is described at the end of the routine description.

3. Cycle steal for channel adapter

This routine tests the outbound cycle steal for the selected CA (data transfer from the CCU storage into the CA buffer).

To test the cycle steal the CE should first reply to the following message:

'TO TEST CYCLE STL TYPE: CE THEN PRESS SEND'."

Followed by:

'ENTER OPT-CA NBR-ADDR-DATA THEN PRESS SEND'.

The CE should enter the parameters in the following format:

'R00xxaaaaaaaddddddddddddddddd' where

00 stands for the option (01, 02, 03 or 06)
xx stands for the CA number under test
a..a stands for the ADDRESS
d..d stands for the DATA

Option 01 A single run of cycle steal with error reporting and return to enter a new request.

Option 02 Loops on the cycle steal until an error is found and reported.

Option 03 Loops on the cycle steal without error reporting.

Option 06 Loop on the cycle steal with data compare until an error is found and reported.

CA NBR is the selected CA to test (range is from 01 to 06).

ADDR is the storage address from which the data is to be cycle stolen. It must be greater than X'008000'.

DATA is 16 hexadecimal digits (8 bytes of data) to cycle steal from CCU storage into the CA buffer.

Error reporting is described at the end of routine description.

MB01

ERC	Function	Error Description	RAC	Comments
7FF0	Process PIO scoping on selected adapter (TA value, R/W) and error reporting. Requested option is: 01	Level 1 interrupt occurred for PIO	675	ERR bits is the image of In X'76'
7FF1	Process PIO scoping on selected adapter (TA value, R/W) and error reporting. Requested option is: 02	Level 1 interrupt occurred for PIO	675	ERR bits is the image of In X'76'
7FFE	Enable redrive to test autoselection mechanism	Error occurred when enabling the redrives	675	ERR bits is the image of In X'76'
7FFA	Test autoselect mechanism: PRI on CA1, second CA not given. Check the CA which has presented level 3 first. Only one CA has been put in the request	Wrong CA number presented its level 3 interrupt when enabling auto-select	675	
7FFB	Test autoselect mechanism: PRI on CA1, PRI on CA2 and check if first level 3 raised by expected CA.	Wrong CA raised level 3 first	675	ERR bits gives X'XXFF' where XX is CA address
7FFC	Check if second level 3 is presented by expected CA.	Wrong CA raised level 3	675	X'XXFF' where XX is CA address
7FFD	Level 1 interrupt processing	A level one occurred during the test of autoselect.	675	ERR bits gives In X'76'
7F00	Enable redrives to test the CA cycle steal.	Error occurred when enabling the redrives.	675	ERR bits gives
7F22	Test CA cycle steal. Set up all parameters to do the cycle steal: - move data into specified storage address - Set up CS pointer - Set outbound	Error occurred in data compare (option 06).	675	ERR bits gives data in error
7F11	Level 1 processing for CS.	Level 1 interrupt occurred during CS process	675	ERR bits gives In X'76'

MB01 - LEVEL 1 INTERRUPT HANDLER

The level 1 interrupt handler processes two types of interrupt:

- IOC level 1 interrupt
- Adapter level 1 interrupt

An adapter level 1 interrupt can be originated by CSPs, TRAs, and by CAs. When originated from the CSP/TRAs while L1 scoping is in progress, the interrupt handler issues a 'Get Error Status' to clear the interrupt. All others cases of L1 interrupts are processed as an error.

If the L1 interrupt is due to none of those mentioned above an ABEND 670 is issued and terminates the routine.

ERC	Function	Error Description	RAC	Comments
7FF9	Reset level 1 interrupt either IOC or adapter	Cannot reset level 1	675	

MB01 - LEVEL 2 INTERRUPT HANDLER

The level 2 interrupt handler processes interrupt requests coming from the CSPs. When L2 scoping is in progress, the interrupt handler issues a 'Get Line ID' to clear the interrupt.

In all other cases, the interrupt handler issues an ABEND 671 and terminates the routine.

ERC	Function	Error Description	RAC	Comments
7FF9	Process level 2 interrupt from CSP/TRA and issue a 'Get Line ID' to clear interrupt.	Cannot reset level 2.	675	

MB01 - LEVEL 3 INTERRUPT HANDLER

The level 3 interrupt handler processes two types of interrupt:

1. Timer interrupt.

   The interrupt is reset and L3 exited.

2. CA level 3 interrupt.

   The interrupt handler saves the CA information (In X'6F') for use by the scoping routine, resets D/S, and exits the routine.

   If an error occurs, the CA is reset and exits the routine.

   In all other cases, an ABEND 670 is requested and the routine terminated.

ERC	Function	Error Description	RAC	Comments
7FE1	Process level 3 interrupts raised by CAs while the autoselect scoping is in progress.	Permanent level 3 interrupt	676	ERR bits gives 1111
7FE2	Process level 1 interrupts when PIO issued from level 3	An IOC level 1 occurred.	676	ERR bits gives In X'76'

CHAPTER 4. CA DIAGNOSTICS

Table of Contents for Chapter 4

CHAPTER 4. CA DIAGNOSTICS

The CA diagnostic group has only one IFT: CA (IFT L).

REQUIREMENTS

Before running the CA diagnostic group you must ensure that the CCU and the IOC bus groups run without error. If not, the results given by the CA diagnostic group may be of no value, or misleading.

SELECTION

For running offline diagnostics, see Chapter 1.

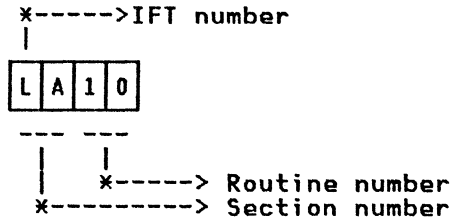
DIAG==>_

4	CA group selected
X	Specific IFT X in this group
XY	Specific section y in IFT X
XYZZ	Specific routine ZZ in section XY

For specific section and routine selection, see routine lists on following pages, and in Chapter 1.

Move the cursor from its initial position (DIAG==>) to the next after each parameter is entered. To skip a parameter entry, press the --> key.

To interpret correctly the results of a selected section or routine, make sure the preceding routines in the group are running without error. The routine identification contains the IFT number, the section number, and the routine number as follows:



ADP#==>_

Enter the selection CA number in the range 1 to 6.

If you do not specify a particular CA, the CA group tests in turn every CA and TPS present in the 3725/3726 CDF and disabled on the control panel (ENBL/DSBL switch of the selected channel adapter interface set to DSBL).

LINE==>_ (not applicable)

OPT==> For option display and description, see Chapter 1.

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK


D:DIAGNOSTICS                                  SP:CCU STOP     Q:DATE/TIME
                                 U:UTILITY PGM ST:CCU START    T:TERMINATE
E:ERROR LOG                                    RS: CCU RESET

DIAG   ADP#  LINE

1 ALL
2 CCU
3 IOCB
4 CA    1-> 6
5 TSS   1->16 0->31
6 OLT   1-> 6
7 TRSS 1->16 1->4

       AND                                          DIAG - RUN INIT

OPT = Y IF MODIFY
OPTION REQUIRED
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> LA10  ADP#==> 2   LINE==>     OPT==> N
===>
```

On the above screen, routine LA10 will run on CA number 2.

Press SEND to execute the request.

Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

MANUAL INTERVENTION ROUTINES

None

IFT DESCRIPTION

The command processor (CP) and the channel adapter group are loaded in the CCU; the DCM takes control in the MOSS. A channel adapter is completely checked out when both the channel adapter group, and the host OLTs run free of errors.

A trace can be taken of all the events during CAIFT run. Refer to 'CAIFT Event Trace' later in this chapter for details.

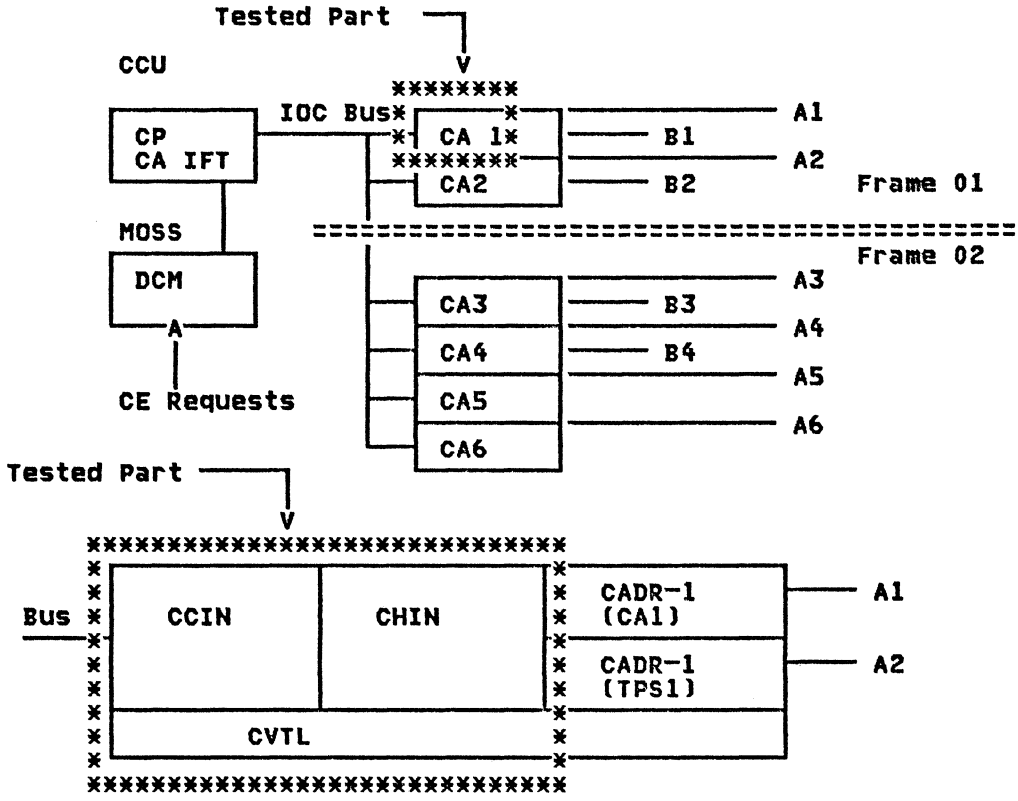


Figure 4-1. CA Diagnostic Testing Sequence

CARD CHANGE

Warning: Do not pull out a CADR card, even if the communication controller is powered off, unless you are sure that the host system is not using this channel interface.

If the host system is using the channel interface, refer to the CADR replacement procedure in Chapter 5 before pulling out the CADR card.

RUNNING TIME

The following running times apply to a single channel only.

Run Init LA	1 min 10 sec (Note 1) Note 2

Notes:

1. Run Init is 13 sec if the command processor (CP) has already been loaded in the CCU by a previous IOC bus or CA run request.
2. 32 times the number of installed CAs (time in seconds).

Messages

None

ADDITIONAL INFORMATION

If an unexpected CCU level 1 interrupt occurs (RACs 690 and 692) while a channel adapter is under test, the error cannot be isolated using the channel adapter diagnostics. The 'ADDIT INFO' field provides information for further troubleshooting.

1. **Adapter Command** (0870 in the example below)

   These bytes indicate which command was sent to the channel adapter or RDV at TA time. For details, refer to Chapter 12.

   The adapter command is the contents of the register specified by R2 in the IOH instruction.

2. **Data** (2100 in the example below)

   These bytes indicate which data was sent at TD time. They are valid for the write command. They are not used for read commands, and are set to FFFF.

3. **CCU level 1 Register (X'7E')** (0100 in the example below)

   This register holds the CCU error. For details, see page 10-230 of the MIM Part 1.

4. **In X'76' contents.** (5800 in the example below)

   This register holds the IOC level 1 error. For details, see page 10-230 of the MIM Part 1.

5. **CAx (for RAC 692 only)** (CAx in the example below)

   Gives the number of the faulty CA.

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
RUN       BYP-ADP-CHK


D:DIAGNOSTICS                                    SP:CCU STOP    Q:DATE/TIME
                                U:UTILITY PGM    ST:CCU START   T:TERMINATE
E:ERROR LOG                                      RS:CCU RESET

                  |
                  | ********************* ORIGIN: CCU <-IFT
                  | * RAC 690      * LEVEL :   X'48'
                  | *     692      * LVLMSK:   X'00'
                  | *               * ROUTINE LA01
                  | ********************* ADDIT INFO:
PANEL  MAINT.     |                     0870  2100  0100
 IML MANDATORY    |                     5800  CA1
                  | START 01:25:52 1ST STOP 01:27:03
                  | REQUEST: LA                      CA   DIAG UNXPTD.ERR
                  | OPTIONS: S   NW C1   R1               ROUTINE LA01 ADP 01
                  |
                  | UNEXPECTED ERROR
T COMMAND BARRED  |
CCU FNCTN BARRED  |
===>
```

CA IFT EVENT TRACE

All events in CA IFT are traced in a fixed area in CCU storage (X'10000' through X'20000').

These events are:

- Type 1: Channel adapter input and check input.
- Type 2: After an expected level 1 interrupt, read the level 1 buffer and check that it contains the expected value.
- Type 3: CCU register input and check value.
- Type 4: Channel adapter command. All events are traced as 8 halfwords; the meaning of each halfword is given in the following table:

	Event Type			
Halfword	1	2	3	4
1	Event type	Event type	Event type	Event type
2	CA input control halfword (Note 1)	Level 1 buffer address (Note 2)	CCU register address	CA command Control word (Note 1)
3	Not used	Not used	Not used	CA command
4	Received data	Received data	Received data	Not used
5	Mask	Mask	Mask	Not used
6	Expected data under mask	Expected data under mask	Expected data under mask	Not used
7	ERC	ERC	ERC	Not used
8	RAC	RAC	RAC	Not used

Notes:

1. The CA input or output control halfword format is X'08XY' where:

   08 is the command sent to the selected CA
   X is the register selected
   Y is 0 (= output) or 1 (= input)

   Example: X'0860' = output X'6', X'0871' = input X'7'.

2. When an expected level 1 interrupt occurs, all results are logged into a fixed table in storage. The addresses and contents are given in the following table:

Address	Contents
X'10EA'	Level 1 switches (see ERC description for meaning)
X'10EC'	CCU register X'7E' at level 1 entry
X'10EE'	CA register X' E' at level 1 entry
X'10F0'	CA register X' D' at level 1 entry
X'10F2'	CCU register X'7E' after reset
X'10F4'	CA register X' E' after reset
X'10F6'	CA register X' D' after reset

HOW TO USE THE TRACE FACILITY

When an error occurs:

1. Note the routine number and do a terminate command or a new IPL to reinitialize the CCU, and to clear CCU storage.

2. Call the diagnostic, then run only the failing routine on the faulty adapter.

3. When the fault is reproduced, use CCU services with option 4 to display location X'10000'. This is the beginning of the trace record. Using the ERC description documentation, follow the normal routine flow and locate the first anomaly. The last event pointer is in CCU location X'1090'.

ROUTINES DESCRIPTION

LA01 - ADAPTER SELECTION TEST

1. Enable redrive specific (redrive address extracted from CA address)
2. Select CA under test (Output X'67' with: CA address, select bit, disable auto select).
3. Input X'6E' to check correct selection
4. Input X'6F' to check correct selection
5. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	3	Input X'6E' under mask X'0E00', expected result X'0000' (notes 1 and 2)
0002	680	4	Input X'6F' under mask X'0E00', expected result X'0000' (notes 1 and 2)
Notes (1) 'ERR BIT' field gives erroneous received pattern (2) X=CA under test from X'0000' to X'1010' (the CA number is given by the ADDR field)			

LA02 - CHANNEL ADAPTER DISABLE TEST

1. Enable redrive specific (redrive address extracted from CA address)
2. Select CA under test (Output X'67' with: CA address, select bit, disable auto select, disable CA interface).
3. Input X'67' to check CA interface is disabled
4. Input X'6E' to check correct selection
5. Input X'6F' to check correct selection
6. Disable redrives broadcast

ERC	RAC	Step	Error Description
2001	681	3	CA under test found enabled after an Out X'67' to disable interface (see note 1)
2002	680	4	Input X'6E' under mask X'0E00', expected result X'0000' (ERR BIT field gives erroneous bits of X (see note 2)
2003	680	5	Input X'6F' under mask X'0E00', expected result X'0000' (ERR BIT field gives erroneous bits of X (see note 2)
Notes (1) 'ERR BIT' field gives erroneous received bit under mask (2) X=CA under test from X'0000' to X'1010' (the CA number is given by the ADDR field)			

LA03 - CHANNEL ADAPTER DIAGNOSTIC RESET

1. Enable redrive specific (redrive address extracted from CA address)
2. Select CA under test (Output X'67' with: CA address, select bit, disable auto select, disable CA interface).
3. Input X'67' to check CA interface is disabled
4. Input X'6E' to check correct CA selection
5. Input X'6F' to check correct CA selection
6. Output X'6C' with pattern X'FF00' to prepare reset test
7. Reset CA (Output X'67' with CA address, select bit, reset bit, disable auto select)
8. Select CA (Output X'67' with CA address, select bit, disable auto select).
9. Input X'67' to check CA interface is always disabled
10. Input X'6E' to check CA is always selected
11. Input X'6F' to check CA is always selected
12. Input X'60' to check register correct reset
13. Input X'62' to check register correct reset
14. Input X'66' to check register correct reset
15. Input X'6C' to check register correct reset
16. Input X'6D' to check register correct reset
17. Input X'6E' to check register correct reset
18. Input X'6F' to check register correct reset
19. Input CCU level 1 register X'7E'
20. Input CCU level 3 register X'77'
21. Disable redrives broadcast

ERC	RAC	Step	Error Description
2001	681	3	CA under test found enabled after an Out X'67' to disable interface ('ERR BIT' = erroneous received bit under mask)
2002	680	4	Input X'6E' under mask X'0E00', expected result X'0X00' ('ERR BIT' = erroneous bits of X) (see note)
2003	680	5	Input X'6F' under mask X'0E00', expected result X'0X00' ('ERR BIT' = erroneous bits of X) (see note)
4301	681	9	CA under test was found enabled after a reset command ('ERR BIT' = erroneous received bit under mask)
4302	681	10	Input X'6E' under mask X'0E00', expected result X'0X00' ('ERR BIT' = erroneous bits of X) (see note)
4303	680	11	Input X'6F' under mask X'0E00', expected result X'0X00' ('ERR BIT' = erroneous bits of X) (see note)
4304	681	12	Input X'60' under mask X'FFFF', expected result X'0000' ('ERR BIT' = bits not reset in register X'60')
4305	681	13	Input X'62' under mask X'FFDF' expected result X'0X00' ('ERR BIT' = bits not reset in register X'62')
4306	681	14	Input X'66' under mask X'FF00', expected result X'0000' ('ERR BIT' = bits not reset in register X'66')
4308	680	15	Input X'6C' under mask X'FFFF', expected result X'00FF' ('ERR BIT' = bits not reset in register X'6C')
4309	680	16	Input X'6D' under mask X'FFFF', expected result X'0000' ('ERR BIT' = bits not reset in register X'6D')
430A	680	17	Input X'6E' under mask X'FFFF', expected result X'0000' ('ERR BIT' = bits not reset in register X'6E')
430B	680	18	Input X'6F' under mask X'31FF', expected result X'0000' ('ERR BIT' = bits not reset in register X'6F')
430C	680	19	CCU register Input X'7E' under mask X'0500', expected result X'0000' ('ERR BIT' = bits not reset in register X'7E')
430D	680	20	CCU register Input X'77' under mask X'0080', expected result X'0000' ('ERR BIT' = bits not reset in register X'7E')

Note: X=CA under test from X'0000' to X'1010' (the CA number is given by the ADDR field

LA04 - RESIDUAL BYTE COUNT

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'6C' with a pattern given in annexed table
4. Input X'6C' to check received pattern equal to pattern sent
5. Return on item 3 until all patterns are sent
6. Output X'6C' with a pattern given in annexed table
7. Input X'6C' to check received pattern equal to pattern sent
8. Reset CA command (Output X'67' with CA address, select bit, reset bit, disable autoselect)
9. Select CA command (Output X'67' with CA address, select bit, disable autoselect)
10. Input X'6C' to check correct reset
11. Return to step 6 until all patterns are sent
12. CA diagnostic reset
13. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	4	Received Input X'6C' does not match pattern sent by Output X'6C' [1]
0002	680	7	Received Input X'6C' does not match pattern sent by Output X'6C' [1]
0003	680	10	Received Input X'6C' after a reset command does not match expected X'00FF' pattern [1]
[1] 'ERR BIT' field gives erroneous bits of received pattern.			

Patterns sent - used in routine LA04:

X'00FA', X'005F', X'00F5', X'0005', X'00AF', X'00A0', X'00A0', X'00A5', X'005A', X'00AA', X'0055', X'0000', X'00FF'.

LA05 - CHARACTER MONITOR SWITCHES

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'6C' with a pattern given in annexed table (see LA04)
4. Input X'6C' under mask X'F000' to check received pattern equal to sent pattern
5. Output X'6C' with data pattern = 0
6. Input X'6C' to check correct reset
7. Return to 3 until all patterns are sent
8. Output X'6C' with a pattern given in annexed table
9. Input X'6C' under mask X'F000' to check received pattern equal to sent pattern
10. Output X'67' (with CA select, CA address, CA reset, disable autoselect) to reset CA.
11. Output X'67' (with CA select, CA address, CA reset, disable autoselect) to again select CA
12. Input X'6C' to check correct register X'6C' reset
13. Return to step 8 until all patterns are sent
14. CA diagnostic reset
15. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	681	4	Received Input X'6C' does not match pattern sent Output X'6C' [1]
0002	681	6	Received Input X'6C' is not zero after Output X'6C' to clear it [1]
0003	681	9	Received Input X'6C' does not match pattern sent [1]
0004	681	12	Received Input X'6C' is not equal to X'00FF' after a reset CA command [1]
[1] 'ERR BIT' field gives erroneous bits of received pattern			

Patterns sent - used in routine LA05:

X'00FA', X'005F', X'00F5', X'0005', X'00AF', X'00A0', X'00A0', X'00A5', X'005A', X'00AA', X'0055', X'0000', X'00FF'.

LA06 - LOCAL STORE TEST

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'66' with pattern X'00A5'
4. Input X'66' to check correct pattern
5. Output X'63' with pattern X'05A0'
6. Input X'63' to check correct pattern
7. Output X'64' with pattern X'50A0'
8. Input X'64' to check correct pattern
9. Output X'65' with pattern X'F0AF'
10. Input X'65' to check correct pattern
11. Output X'66' with pattern X'10A5' to set diagnostic memory mode
12. Output X'64' (buffer A bytes 4-5) with pattern X'A5A5'
13. Input X'64' (buffer A bytes 4-5) to check correct pattern
14. Output X'65' (buffer A bytes 6-7) with pattern X'05A0'
15. Input X'65' (buffer A bytes 6-7) to check correct pattern
16. Output X'66' with pattern X'00A5' to reset diagnostic memory mode
17. Input X'66' to check X'00A5'
18. Input X'63' to check X'05A0'
19. Input X'64' to check X'50A0'
20. Input X'65' to check X'F0AF'
21. Rerun steps 3 through 20 with following patterns:
    X'66'=X'005A', X'63'=X'A5A5', X'64'=X'05A0, X'65'=X'50A0',
    X'64'E=X'5A5A', X'65'E=X'A5A5'

    Error codes are in the range X'0100'
22. Rerun steps 3 through 20 with following patterns:
    X'66'=X'00AA', X'63'=X'A5A5', X'64'=X'A5A5', X'65'=X'50A0',
    X'64'E=X'55AA', X'65'E=X'5A5A'

    Error codes are in the range X'0200'
23. Rerun steps 3 through 20 with following patterns:
    X'66'=X'0055', X'63'=X'55AA', X'64'=X'5A5A', X'65'=X'A5A5',
    X'64'E=X'AA55', X'65'E=X'55AA'

    Error codes are in the range X'0300'
24. Rerun steps 3 through 20 with following patterns:
    X'66'=X'0000', X'63'=X'AA55', X'64'=X'55AA', X'65'=X'5A5A',
    X'64'E=X'0000', X'65'E=X'AA55'

    Error codes are in the range X'0400'
25. CA diagnostic reset
26. Disable redrives broadcast

(LA06)

ERC	RAC	Step	Error Description
0016	680	4	Received Input X'66' does not match expected X'00A5' pattern
0013	680	6	Received Input X'63' does not match expected X'05A0' pattern
0014	680	8	Received Input X'64' does not match expected X'50A0' pattern
0015	680	10	Received Input X'65' does not match expected X'F0AF' pattern
0024	680	13	Received buffer A bytes 4-5 do not match expected X'A5A5' pattern
0025	680	15	Received buffer A bytes 6-7 do not match expected X'05A0' pattern
0036	680	17	Received Input X'66' does not match expected X'00A5' pattern
0033	680	18	Received Input X'63' does not match expected X'05A0' pattern
0034	680	19	Received Input X'64' does not match expected X'50A0' pattern
0035	680	20	Received Input X'65' does not match expected X'F0AF' pattern
0116	680	4,21	Received Input X'66' does not match expected X'005A' pattern
0113	680	6,21	Received Input X'63' does not match expected X'A5A5' pattern
0114	680	8,21	Received Input X'64' does not match expected X'05A0' pattern
0115	680	10,21	Received Input X'65' does not match expected X'50A0' pattern
0124	680	13,21	Received buffer A bytes 4-5 do not match expected X'5A5A' pattern
0125	680	15,21	Received buffer A bytes 6-7 do not match expected X'A5A5' pattern
0136	680	17,21	Received Input X'66' does not match expected X'005A' pattern
0133	680	18,21	Received Input X'63' does not match expected X'A5A5' pattern
0134	680	19,21	Received Input X'64' does not match expected X'05A0' pattern
0135	680	20,21	Received Input X'65' does not match expected X'50A0' pattern
0216	680	4,22	Received Input X'66' does not match expected X'00AA' pattern
0213	680	6,22	Received Input X'63' does not match expected X'5A5A' pattern
0214	680	8,22	Received Input X'64' does not match expected X'A5A5' pattern
0215	680	10,22	Received Input X'65' does not match expected X'05A0' pattern
0224	680	13,22	Received buffer A bytes 4-5 does not match expected X'55AA' pattern
0225	680	15,22	Received buffer A bytes 6-7 does not match expected X'5A5A' pattern
0236	680	17,22	Received Input X'66' does not match expected X'00AA' pattern
0233	680	18,22	Received Input X'63' does not match expected X'5A5A' pattern
0234	680	19,22	Received Input X'64' does not match expected X'A5A5' pattern
0235	680	20,22	Received Input X'65' does not match expected X'05A0' pattern
0316	680	4,23	Received Input X'66' does not match expected X'0055' pattern
0313	680	6,23	Received Input X'63' does not match expected X'55AA' pattern
0314	680	8,23	Received Input X'64' does not match expected X'5A5A' pattern
0315	680	8,23	Received Input X'65' does not match expected X'A5A5' pattern
0324	680	13,23	Received buffer A bytes 4-5 does not match expected X'AA55' pattern
0325	680	15,23	Received buffer A bytes 6-7 does not match expected X'55AA' pattern
0336	680	17,23	Received Input X'66' does not match expected X'0055' pattern
0333	680	18,23	Received Input X'63' does not match expected X'55AA' pattern
0334	680	19,23	Received Input X'64' does not match expected X'5A5A' pattern
0335	680	20,23	Received Input X'65' does not match expected X'A5A5' pattern
0416	680	4,24	Received Input X'66' does not match expected X'0000' pattern
0413	680	6,24	Received Input X'63' does not match expected X'AA55' pattern
0414	680	8,24	Received Input X'64' does not match expected X'55AA' pattern
0415	680	10,24	Received Input X'65' does not match expected X'5A5A' pattern
0424	680	13,24	Received buffer A bytes 4-5 does not match expected X'0000' pattern
0425	680	15,24	Received buffer A bytes 6-7 does not match expected X'AA55' pattern
0436	680	17,24	Received Input X'66' does not match expected X'0000' pattern
0433	680	18,24	Received Input X'63' does not match expected X'AA55' pattern
0434	680	19,24	Received Input X'64' does not match expected X'55AA' pattern
0435	680	20,24	Received Input X'65' does not match expected X'5A5A' pattern

LA07 - PIO MODE

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Set PIO mode (Output X'62' with PIO mode pattern X'0800')
4. Read PIO mode in register X'67' (pattern X'0200')
5. Reset PIO mode (Output X'62' with pattern X'0000')
6. Check correct PIO mode reset in register X'67'
7. Set PIO mode again (Output X'62' with PIO mode pattern X'0800')
8. Read PIO mode in register X'67' (pattern X'0200')
9. Reset channel adapter (Output X'67' with CA address, select bit, reset bit, disable auto select)
10. Select channel adapter (Output X'67' with CA address, select bit, disable auto select)
11. Check correct PIO mode reset in register X'67'
12. CA diagnostic reset
13. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	681	4	PIO mode bit is not set in Received Input X'67' [1]
0002	681	6	PIO mode bit is not reset in Received Input X'67' [1]
0003	681	8	PIO mode bit is not set in Received Input X'67' [1]
0004	681	11	PIO mode bit is not reset in Received Input X'67' after a reset CA command [1]

[1] 'ERR BIT' = X'0200' (PIO mode in register X'67')

LA08 - TRANSFER SEQUENCE SWITCHES

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Set program request interrupt (Output X'67' with CA address, select bit, Program request interrupt bit)
4. Input CCU register X'77' to check CA level 3 interrupt request
5. Input X'6F' to check data status level 3 bit
6. Input X'62' to check program request bit in register X'62'
7. Reset data status level 3 interrupt (Output X'62' with pattern X'0200')
8. Input X'62' to check correct reset
9. Input X'6F' to check correct reset
10. Input CCU register X'77' to check reset of CA level 3 request bit
11. Reset cycle steal count to avoid starting erroneous AIO operations
12. Output X'62' with pattern given in annexed table
13. Input X'62' to check under mask correct register X'62' setting
14. Reset channel adapter (Output X'67' with CA address, select bit, reset bit, disable auto select)
15. Select channel adapter (Output X'67' with CA address, select bit, disable auto select)
16. Input X'62' to check correct reset
17. Input X'67' to check correct reset
18. Change pattern used; return to step 11 until end of table is reached
19. Output X'67' with CA select, CA address, and program request interrupt
20. Input X'62' to check under mask program request interrupt bit
21. Input CCU register X'77' to check CA level 3 request bit
22. Input X'6F' to check data status level 3 request
23. Output X'62' with pattern given in annexed table
24. Input X'62' to check under mask correct register X'62' setting
25. CA diagnostic reset
26. Disable redrives broadcast

ERC	RAC	Step	Error Description
5A01	680	4	No CA level 3 interrupt request after a set of PRI bit
5A02	680	5	Data status level 3 bit not present in Received Input X'6F'
5A03	681	6	No program request interrupt bit in Received Input X'62'
5A04	680	8	Received Input X'6F' does not match expected pattern mask X'FFDF' [1]
5A05	680	9	Received Input X'62' does not match expected pattern mask X'31FF' [1]
5A06	681	10	L3 request bit not reset in CCU register X'77'
0001	681	13	Received Input X'62' does not match expected pattern mask X'F000', expected pattern given below
0002	681	16	Received Input X'62' does not match expected pattern mask X'FFDF' [1]
0003	681	17	Received Input X'67' does not match expected pattern mask X'FFFF' [1]
0004	681	20	Program request interrupt is not set in Received Input X'62'
0005	681	21	CA level 3 bit is not set in CCU register X'77'
0006	681	22	Data status level 3 bit not set in Received Input '6F'
0007	681	23	Received Input X'62' does not match expected pattern mask X'F000' [1]

[1] 'ERR BIT' = erroneous bits of received pattern under mask

Patterns sent:

X'F800', X'E800', X'D800', X'C800', X'B800', X'A800', X'9800', X'8800', X'7800', X'6800', X'5800', X'4800', X'3800', X'2800', X'1800'.

LA09 - FORCE BUSY

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'66' to reset CA to neutral state
4. Output X'66' to set (that is, to force) A busy
5. Input X'66' to check A busy correct setting
6. Output X'66' with reset CA to neutral state
7. Input X'66' to check A busy correct reset
8. Output X'66' to force A busy
9. Input X'66' to check A busy correct setting
10. Input X'6F' to check if TPS is installed
11. If not installed, go to step 18
12. Output X'66' to force B busy
13. Input X'66' to check B busy correct setting
14. Output X'66' to reset CA to neutral state
15. Input X'66' to check B busy correct reset
16. Output X'66' to force A busy
17. Input X'66' to check A busy correct setting
18. CA diagnostic reset
19. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	681	5	Check for A busy correct setting in Received Input X'66' [1]
0002	681	7	Check for A busy correct reset after reset to neutral state in Received Input X'66' [1]
0003	681	9	Check for A busy correct setting in Received Input X'66' [1]
0004	681	13	Check for B busy correct setting in Received Input X'66' [1]
0005	680	15	Check for B busy correct reset after reset to neutral neutral state in Received Input X'66' [1]
0006	681	17	Check for B busy correct setting in Received Input X'66' [1]

[1] 'ERR BIT' = erroneous bit of received pattern

LA0A - PROGRAM REQUEST INTERRUPT

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'67' to set PRI (CA address, select CA, disable auto select, set PRI)
4. Software timer to wait for interrupt
5. Input X'77' to check CA level 3 request
6. Input X'6F' to check data status level 3 request
7. Input X'62' to check program request interrupt
8. Output X'62' to reset data status level 3 request
9. Input X'62' to check correct reset
10. Input X'6F' to check data status reset
11. Input X'77' to check reset of CA level 3 request
12. Output X'67' to again set PRI (CA address, CA select, disable auto select, set PRI)
13. Software timer to wait for interrupt level 3
14. Input X'77' to check CA level 3 request
15. Input X'6F' to check data status level 3 request
16. Input X'62' to check program request interrupt
17. CA diagnostic reset
18. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	5	No CA level 3 request in Received Input X'77'
0002	680	6	No data status level request in Received Input X'6F'
0003	681	7	No PRI bit in Received Input X'62'
0004	680	9	Received Input X'62'after data status reset, is not equal to X'0007' ('ERR BIT' = erroneous bits of Received Input X'62')
0005	680	10	Level 3 request not reset in Received Input X'6F' ('ERR BIT' = erroneous bits of Received Input X'6F')
0006	680	11	CA level 3 request not reset in Received Input X'77'
0007	680	14	No CA level 3 request in Received Input X'77'
0008	680	15	No data status level 3 request in Received Input X'6F'
0009	681	16	No PRI bit in Received Input X'62'

LA0B - SUPPRESS OUT MONITOR INTERRUPT

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'67' to set suppress out monitor bit (CA address, CA select, disable auto select, set suppress out monitor)
4. Input X'62' to check suppress out monitor interrupt
5. Input X'6F' to check data status level 3 interrupt
6. Input X'77' to check CA level 3 request
7. Output X'62' to reset data status level 3 request
8. Input X'62' to check correct reset
9. Input X'6F' to check reset of data status level 3 request
10. Input X'77' to check reset of CA level 3 request
11. Output X'67' to again set suppress out monitor (CA address, CA select, disable autoselect, set suppress out monitor)
12. Input X'62' to check suppress out monitor interrupt
13. Input X'6F' to check data status level 3 request
14. Input X'77' to check CA level 3 request
15. CA diagnostic reset
16. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	4	No suppress out monitor interrupt in Received Input X'62'
0002	680	5	No data status level 3 in Received Input X'6F'
0003	681	6	No CA level 3 request in Received Input X'77'
0004	680	8	Unexpected pattern in Received Input X'62' after data status reset ('ERR BIT' = erroneous received bits)
0005	680	9	Unexpected pattern in Received Input X'6F'
0006	680	10	CA level 3 request not reset in Received Input X'77'
0007	680	12	No suppress out monitor interrupt in Received Input X'62'
0008	680	13	No data status level 3 in Received Input X'6F'
0009	681	14	No CA level 3 in Received Input X'77'

LA0C - COMMANDS DECODE CHECK

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Send Output X'68' and wait for level 1 interrupt
4. Check if a level 1 interrupt occurred
5. Input X'7E' to check adapter level 1 request
6. Input X'6E' to check tested adapter level 1 request
7. Input X'6D' to check CCU interconnection card check
8. Reset level 1 request (Output X'67' with reset level 1 checks)
9. Input X'7E' to check reset of adapter level 1 request
10. Input X'6E' to check correct reset
11. Input X'6D' to check correct reset
12. Send Output X'68' and Wait for level 1 interrupt
13. Check if a level 1 interrupt occurred
14. Input X'7E' to check adapter level 1 request
15. Input X'6E' to check tested adapter level 1 request
16. Input X'6D' to check CCU interconnection card check
17. Input X'67' to check correct reset
18. Reset level 1 request (Output X'67' with reset CA)
19. Input X'7E' to check reset of adapter level 1 request
20. Input X'6E' to check correct reset
21. Input X'6D' to check reset of CCU interconnection card check
22. -Change Output X'68' to Output X'69' (in step 3)

    - Error codes X'0001' through X'000E' are changed to X'000F' through X'001C'

    - Return to step 3 and rerun steps 3 through 20

23. -Change Output X'69' to Output X'6A' (in step 3)

    - Error codes X'000F' through X'001C' are changed to X'001D' through X'002A'

    - Return to step 3 and rerun steps 3 through 20

24. CA diagnostic reset
25. Disable redrives broadcast

(LA0C)

ERC	RAC	Step	Error Description
0001	680	4	No level 1 occurred after Output X'68'
0002	680	5	No adapter level 1 request in Received Input X'7E'
0003	680	6	No level 1 request, or level 1 from another CA than the tested one, in Received Input X'6E'
0004	680	7	No CCU interconnection card check in Received Input X'6D'
0005	680	9	Adapter level 1 request not reset in Received Input X'7E' [1]
0006	680	10	CA level 1 request not reset in Received Input X'6E' [1]
0007	680	11	Received Input X'6D' not reset [1]
0008	680	13	No level 1 occurred after output X'68'
0009	680	14	No adapter level 1 request in Received Input X'7E'
000A	680	15	No level 1 request, or level 1 from another CA than the tested one, in Received Input X'6E'
000B	680	16	No CCU interconnection card check in Received Input X'6D'
000C	680	18	Adapter level 1 request not reset in Received Input X'7E' [1]
000D	680	19	CA level 1 request not reset in Received Input X'6E' [1]
000E	680	20	Received Input X'6D' not reset [1]
000F	680	4,21	No level 1 occurred after Output X'69'
0010	680	5,21	No adapter level 1 request in Received Input X'7E'
0011	680	6,21	No level 1 request, or level 1 from another CA than the tested one, in Received Input X'6E'
0012	680	7,21	No CCU interconnection card check in Received Input X'6D'
0013	680	9,21	Adapter level 1 request not reset in Received Input X'7E' [1]
0014	680	10,21	CA level 1 request not reset in Received Input X'6E' [1]
0015	680	11,21	Received Input X'6D' not reset [1]
0016	680	13,21	No level 1 occurred after Output X'69'
0017	680	14,21	No adapter level 1 request in Received Input X'7E'
0018	680	15,21	No level 1 request, or level 1 from another CA than the tested one, in Received Input X'6E'
0019	680	16,21	No CCU interconnection card check in Received Input X'6D'
001A	680	18,21	Adapter level 1 request not reset in Received Input X'7E' [1]
001B	680	19,21	CA level 1 request not reset in Received Input X'6E' [1]
001C	680	20,21	Received Input X'6D' not reset
001D	680	4,22	No level 1 occurred after Output X'6A'
001E	680	5,22	No adapter level 1 request in Received Input X'7E'
001F	680	6,22 2	No level 1 request, or level 1 from another CA than the tested one, in Received Input X'6E'
0020	680	7,22	No CCU interconnection card check in Received Input X'6D'
0021	680	9,22	Adapter level 1 request not reset in Received Input X'7E' [1]
0022	680	10,22	CA level 1 request not reset in Received Input X'6E' [1]
0023	680	11,22	Received Input X'6D' not reset [1]
0024	680	13,22	No level 1 occurred after Output X'6A'
0025	680	14,22	No adapter level 1 request in Received Input X'7E'
0026	680	15,22	No level 1 request, or level 1 from another CA than the tested one, in Received Input X'6E'
0027	680	16,22	No CCU interconnection card check in Received Input X'6D'
0028	680	18,22	Adapter level 1 request not reset in Received Input X'7E' [1]
0029	680	19,22	CA level 1 request not reset in Received Input X'6E' [1]
002A	680	20,22	Received Input X'6D' not reset [1]

[1] 'ERR BIT' = erroneous received bits

LA0D - I/O CHECK AFTER OUTPUT X'6B'

1. Enable redrive specific (redrive address extracted from CA address)
2. CA 680 diagnostic reset, and CA selection
3. Send Output X'6B' and wait for level 1 interrupt
4. Check if a level 1 interrupt occurred
5. Input X'7E' to check level 1 summary
6. Input X'6E' to check no adapter level 1
7. Input X'6D' to check output exception
8. Output X'67' to reset level 1 checks
9. Input X'7E' to check no level 1 request
10. Input X'6E' to check no CA level 1 request
11. Input X'6D' to check correct output exception reset
12. CA diagnostic reset
13. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	4	No level 1 occurred after Output X'6B'
0002	680	5	No level 1 summary in Received Input X'7E'
0003	680	6	level 1 request erroneously set in Received Input X'6E'
0004	680	7	No output exception in Received Input X'6D'
0005	680	9	Always level 1 request after reset level 1 checks [1]
0006	680	10	CA level 1 request set in Received Input X'6E' [1]
0007	680	11	Non-zero pattern in Received Input X'6D' [1]

[1] 'ERR BIT' = erroneous received bits

LA10 - COMPLETE LOCAL STORE TEST

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'67' to set program request interrupt
4. Software timer to wait for interrupt
5. Input X'77' to check CA level 3 request
6. Input X'6F' to check data status level 3
7. Input X'62' to check program request interrupt
8. Output X'62' to reset data status level 3 interrupt
9. Input X'62' to check correct reset
10. Input X'6F' to check = 0
11. Input X'77' to check reset of CA level 3 request
12. Output X'67 to select CA and set PRI bit
13. Software timer to wait for interrupt
14. Input X'77' to check CA level 3 request
15. Input X'6F' to check data/status level 3 request
16. Input X'62' to check program request interrupt
17. Output X'66' with pattern X'00A5'
18. Input X'66' to check received pattern
19. Output X'63' with pattern X'50A0'
20. Input X'63' to check received pattern
21. Output X'64' with pattern X'F0AF'
22. Input X'64' to check received pattern
23. Output X'65' with pattern X'AF05'
24. Input X'65' to check received pattern
25. Output X'61' with pattern X'5FF5'
26. Input X'61' to check received pattern
27. Output X'6B' with pattern X'F55F'
28. Input X'6B' to check received pattern
29. Output X'66' to set diag memory mode
30. Output X'64'E with pattern X'05A0'
31. Input X'64'E to check received pattern
32. Output X'65'E with pattern X'50A0'
33. Input X'65'E to check received pattern
34. Output X'66' to reset diag memory mode
35. Input X'66' to check previous pattern
36. Input X'63' to check previous pattern
37. Input X'64' to check previous pattern
38. Input X'65' to check previous pattern
39. Input X'61' to check previous pattern
40. Input X'6B' to check previous pattern
41. Run again steps 17 through 40 with following patterns:
    X'66'=X'005A', X'63'=X'05A0', X'64'=X'50A0', X'65'=X'F0AF',
    X'61'=X'AF05', X'6B'=X'5FF5', X'64'E=X'A5A5', X'65'E=X'05A0'
    Error codes are changed from 4 through 11 to 12 through 1F
42. Run again steps 17 through 40 with following patterns:
    X'66'=X'00AA', X'63'=X'A5A5', X'64'=X'05A0', X'65'=X'50A0',
    X'61'=X'F0AF', X'6B'=X'AF05', X'64'E=X'5A5A', X'65'E=X'A5A5'
    Error codes are changed from 12 through 1F to 20 through 2D
43. Run again steps 17 through 40 with following patterns:
    X'66'=X'0055', X'63'=X'5A5A', X'64'=X'A5A5', X'65'=X'05A0',
    X'61'=X'50A0', X'6B'=X'F0AF', X'64'E=X'55AA', X'65'E=X'5A5A'
    Error codes are changed from 20 through 2D to 2E through 3B
44. Run again steps 17 through 40 with following patterns:
    X'66'=X'0000', X'63'=X'55AA', X'64'=X'5A5A', X'65'=X'A5A5',
    X'61'=X'05A0', X'6B'=X'50A0', X'64'E=X'AA55', X'65'E=X'55AA'
    Error codes are changed from 2E through 3B to 3C through 49
45. Run again steps 17 through 40 with following patterns:
    X'66'=X'00FF', X'63'=X'AA55', X'64'=X'55AA', X'65'=X'5A5A',
    X'61'=X'A5A5', X'6B'=X'05A0', X'64'E=X'0000', X'65'E=X'AA55'
    Error codes are changed from 3C through 49 to 4A through 57
46. CA diagnostic reset
47. Disable redrives broadcast

(LA10)

ERC	RAC	Step	Error Description
5A01	680	5	No CA level 3 request in Received Input X'77'
5A02	680	6	No data status level request in Received Input X'6F'
5A03	680	7	No program request interrupt in Received Input X'62'
5A04	680	9	Program request interrupt not reset in Received Input X'62'
5A05	680	10	Received Input X'6F' not zero [1]
5A06	680	11	CA level 3 request not reset in Received Input X'77'
0001	680	14	No CA level 3 request in Received Input X'77'
0002	680	15	No data status level 3 in Received Input X'6F'
0003	680	16	No program request interrupt in Received Input X'62'
0004	680	18	Received Input X'66' does not match expected X'00A5' pattern [1]
0005	680	20	Received Input X'63' does not match expected X'50A0' pattern [1]
0006	680	22	Received Input X'64' does not match expected X'F0AF' pattern [1]
0007	680	24	Received Input X'65' does not match expected X'AF05' pattern [1]
0008	680	26	Received Input X'61' does not match expected X'5FF5' pattern [1]
0009	680	28	Received Input X'6B' does not match expected X'F55F' pattern [1]
000A	680	31	Received Input X'64' does not match expected X'05A0' pattern [1]
000B	680	33	Received Input X'65' does not match expected X'50A0' pattern [1]
000C	680	18	Received Input X'66' does not match expected X'00A5' pattern [1]
000D	680	20	Received Input X'63' does not match expected X'50A0' pattern [1]
000E	680	22	Received Input X'64' does not match expected X'F0AF' pattern [1]
000F	680	24	Received Input X'65' does not match expected X'AF05' pattern [1]
0010	680	26	Received Input X'61' does not match expected X'5FF5' pattern [1]
0011	680	28	Received Input X'6B' does not match expected X'F55F' pattern [1]
0012	680	18,41	Received Input X'66' does not match expected X'005A' pattern [1]
0013	680	20,41	Received Input X'63' does not match expected X'05A0' pattern [1]
0014	680	22,41	Received Input X'64' does not match expected X'50A0' pattern [1]
0015	680	24,41	Received Input X'65' does not match expected X'F0AF' pattern [1]
0016	680	26,41	Received Input X'61' does not match expected X'AF05' pattern [1]
0017	680	28,41	Received Input X'6B' does not match expected X'5FF5' pattern [1]
0018	680	31,41	Received Input X'64' does not match expected X'A5A5' pattern [1]
0019	680	33,41	Received Input X'65' does not match expected X'05A0' pattern [1]
001A	680	18,41	Received Input X'66' does not match expected X'005A' pattern [1]
001B	680	20,41	Received Input X'63' does not match expected X'05A0' pattern [1]
001C	680	22,41	Received Input X'64' does not match expected X'50A0' pattern [1]
001D	680	24,41	Received Input X'65' does not match expected X'F0AF' pattern [1]
001E	680	26,41	Received Input X'61' does not match expected X'AF05' pattern [1]
001F	680	28,41	Received Input X'6B' does not match expected X'5FF5' pattern [1]
0020	680	18,42	Received Input X'66' does not match expected X'00AA' pattern [1]
0021	680	20,42	Received Input X'63' does not match expected X'A5A5' pattern [1]
0022	680	22,42	Received Input X'64' does not match expected X'05A0' pattern [1]
0023	680	24,42	Received Input X'65' does not match expected X'50A0' pattern [1]
0024	680	26,42	Received Input X'61' does not match expected X'F0AF' pattern [1]
0025	680	28,42	Received Input X'6B' does not match expected X'AF05' pattern [1]
0026	680	31,42	Received Input X'64' does not match expected X'5A5A' pattern [1]
0027	680	33,42	Received Input X'65' does not match expected X'A5A5' pattern [1]
0028	680	18,42	Received Input X'66' does not match expected X'00AA' pattern [1]
0029	680	20,42	Received Input X'63' does not match expected X'A5A5' pattern [1]
002A	680	22,42	Received Input X'64' does not match expected X'05A0' pattern [1]
002B	680	24,42	Received Input X'65' does not match expected X'50A0' pattern [1]
002C	680	26,42	Received Input X'61' does not match expected X'F0AF' pattern [1]
002D	680	28,42	Received Input X'6B' does not match expected X'AF05' pattern [1]
002E	680	18,43	Received Input X'66' does not match expected X'0055' pattern [1]
002F	680	20,43	Received Input X'63' does not match expected X'5A5A' pattern [1]
0030	680	22,43	Received Input X'64' does not match expected X'A5A5' pattern [1]
0031	680	24,43	Received Input X'65' does not match expected X'05A0' pattern [1]
0032	680	26,43	Received Input X'61' does not match expected X'50A0' pattern [1]
0033	680	28,43	Received Input X'6B' does not match expected X'F0AF' pattern [1]
0034	680	31,43	Received Input X'64' does not match expected X'55AA' pattern [1]
0035	680	33,43	Received Input X'65' does not match expected X'5A5A' pattern [1]
0036	680	18,43	Received Input X'66' does not match expected X'0055' pattern [1]

[1] 'ERR BIT' field gives erroneous received bits

(LA10, continued)

ERC	RAC	Step	Error Description
0037	680	20,43	Received Input X'63' does not match expected X'5A5A' pattern [1]
0038	680	22,43	Received Input X'64' does not match expected X'A5A5' pattern [1]
0039	680	24,43	Received Input X'65' does not match expected X'05A0' pattern [1]
003A	680	26,43	Received Input X'61' does not match expected X'50A0' pattern [1]
003B	680	28,43	Received Input X'6B' does not match expected X'F0AB' pattern [1]
003C	680	18,44	Received Input X'66' does not match expected X'0000' pattern [1]
003D	680	20,44	Received Input X'63' does not match expected X'55AA' pattern [1]
003E	680	22,44	Received Input X'64' does not match expected X'5A5A' pattern [1]
003F	680	24,44	Received Input X'65' does not match expected X'A5A5' pattern [1]
0040	680	26,44	Received Input X'61' does not match expected X'05A0' pattern [1]
0041	680	28,44	Received Input X'6B' does not match expected X'50A0' pattern [1]
0042	680	31,44	Received Input X'64' does not match expected X'AA55' pattern [1]
0043	680	33,44	Received Input X'65' does not match expected X'55AA' pattern [1]
0044	680	18,44	Received Input X'66' does not match expected X'0000' pattern [1]
0045	680	20,44	Received Input X'63' does not match expected X'55AA' pattern [1]
0046	680	22,44	Received Input X'64' does not match expected X'5A5A' pattern [1]
0047	680	24,44	Received Input X'65' does not match expected X'A5A5' pattern [1]
0048	680	26,44	Received Input X'61' does not match expected X'05A0' pattern [1]
0049	680	28,44	Received Input X'6B' does not match expected X'50A0' pattern [1]
004A	680	18,45	Received Input X'66' does not match expected X'00FF' pattern [1]
004B	680	20,45	Received Input X'63' does not match expected X'AA55' pattern [1]
004C	680	22,45	Received Input X'64' does not match expected X'55AA' pattern [1]
004D	680	24,45	Received Input X'65' does not match expected X'5A5A' pattern [1]
004E	680	26,45	Received Input X'61' does not match expected X'A5A5' pattern [1]
004F	680	28,45	Received Input X'6B' does not match expected X'05A0' pattern [1]
0050	680	31,45	Received Input X'64' does not match expected X'0000' pattern [1]
0051	680	33,45	Received Input X'65' does not match expected X'AA55' pattern [1]
0052	680	18,45	Received Input X'66' does not match expected X'00FF' pattern [1]
0053	680	20,45	Received Input X'63' does not match expected X'AA55' pattern [1]
0054	680	22,45	Received Input X'64' does not match expected X'55AA' pattern [1]
0055	680	24,45	Received Input X'65' does not match expected X'5A5A' pattern [1]
0056	680	26,45	Received Input X'61' does not match expected X'A5A5' pattern [1]
0057	680	28,45	Received Input X'6B' does not match expected X'05A0' pattern [1]

[1] 'ERR BIT' field gives erroneous received bits

LA13 - INBOUND PIO BUS PARITY CHECK

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'66' to set check checkers
4. Output X'64' to load register with pattern X'1110' and with a bad parity and ignore the corresponding level 1 result
5. Input X'64' to read register with bad parity, and wait for level 1
6. Check if level 1 occurred
7. Input X'7E' to check adapter level 1 request
8. Input X'6E' to check for any CA level 1 and for level 1 from tested CA
9. Input X'6D' to check internal bus parity error and CCU interconnection card check
10. Reset level 1 checks (Output X'67' with reset level 1 checks bit)
11. Input X'7E' to check no level 1 request from adapter or IOC
12. Input X'6E' to check no level request from CA
13. Input X'6D' to check reset of internal bus parity error and CCU interconnection card check
14. Change reset level 1 checks to reset CA in level 1 treatment
    - Replace error codes X'0001' through X'0007' with X'0008' through X'000E'
    - Rerun steps 3 through 13
15. Change pattern loaded in register X '64' from X'1110' to X'1011'
    - Change reset CA to reset level 1 checks in level 1 treatment
    - Replace error codes X'0008' through X'000E' with X'0011' through X'0017'
    - Rerun steps 3 through 13
16. Change reset level 1 checks to reset CA in level 1 treatment
    - Replace error codes X'0011' through X'0017' with X'0018' through X'001E'
    - Rerun steps 3 through 13
17. CA diagnostic reset
18. Disable redrives broadcast

(LA13)

ERC	RAC	Step	Error Description
0001	680	6	No level 1 occurred after Input X'64'
0002	680	7	No level 1 request from adapter in received Input X'7E'
0003	680	8	No CA level 1 request, or request by a CA other than the tested one, in Received Input X'6E' [1]
0004	680	9	No internal bus parity error or CCU interconnection card check in Received Input X'6D' [1]
0005	680	11	Always level 1 request from adapter or IOC in Received Input X'7E' [1]
0006	680	12	Received Input X'6E' is not reset after a reset level 1 check [1]
0007	680	13	Received Input X'6D' is not reset after a reset level 1 check
0008	680	6,14	No level 1 occurred after Input X'64'
0009	680	7,14	No level 1 request from adapter in received Input X'7E'
000A	680	8,14	No CA level 1 request, or request by a CA other than the tested one, in Received Input X'6E' [1]
000B	680	9,14	No internal bus parity error or CCU interconnection card check in Received Input X'6D' [1]
000C	680	11,14	Always Level 1 request from adapter or IOC in Received Input X'7E' [1]
000D	680	12,14	Received Input X'6E' is not reset after a reset CA [1]
000E	680	13,14	Received Input X'6D' is not reset after a reset CA
0011	680	6,14	No level 1 occurred after Input X'64'
0012	680	7,15	No level 1 request from adapter in received Input X'7E'
0013	680	8,15	No CA level 1 request, or request by a CA other than the tested one, in Received Input X'6E' [1]
0014	680	9,15	No internal bus parity error or CCU interconnection card check in Received Input X'6D' [1]
0015	680	11,15	Always level 1 request from adapter or IOC in Received Input X'7E' [1]
0016	680	12,15	Received Input X'6E' is not reset after a reset level 1 check [1]
0017	680	13,15	Received Input X'6D' is not reset after a reset level 1 check
0018	680	6,16	No level 1 occurred after Input X'64'
0019	680	7,16	No level 1 request from adapter in Received Input X'7E'
001A	680	8,16	No CA level 1 request, or request by a CA other than the tested one, in Received Input X'6E' [1]
001B	680	9,16	No internal bus parity error or CCU interconnection card check in Received Input X'6D' [1]
001C	680	11,16	Always level 1 request from adapter or IOC in Received Input X'7E'
001D	680	12,16	Received Input X'6E' is not reset after a reset CA [1]
001E	680	13,16	Received Input X'6D' is not reset after a reset CA

[1] 'ERR BIT' = erroneous bits received

LA14 - INTERNAL BUS PARITY CHECK

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'66' to set check checkers
4. Output X'62' with pattern X'1888' to get parity error
5. Check that a level 1 occurred
6. Input X'7E' to check level 1 request from adapter
7. Input X'6E' to check for any CA level 1 and for level 1 from tested CA
8. Input X'6D' to check internal bus parity check and CCU interconnection card check
9. Output X'67' to reset level 1 checks
10. Input X'7E' to check no level 1 request from adapter or IOC
11. Input X'6E' to check no level 1 request from CA
12. Input X'6D' to check reset of internal bus parity error and CCU interconnection card check
13. Change data pattern loaded in register X'62' from X'1888' to X'1088'
    - Change error codes X'0001' through X'0007' to X'0008' through X'000E'
    - Rerun steps 3 through 12
14. CA diagnostic reset
15. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	5	No level 1 occurred after X'62' with bad parity forced
0002	680	6	No level 1 request from adapter in received Input X'7E'
0003	680	7	No CA level 1 request, or request from a CA other than the tested one, in Received Input X'6E' [1]
0004	680	8	No internal bus parity check or CCU interconnection card check in Received Input X'6D' [1]
0005	680	10	Level 1 request from adapter or IOC not reset in Received Input X'7E' [1]
0006	680	11	CA level 1 request not reset in Received Input X'6E' [1]
0007	680	12	Internal bus parity check or CCU interconnection card check not reset in Received Input X'6D'
0008	680	5,13	No level 1 occurred after Out X'62' with bad parity forced
0009	680	6,13	No level 1 request from adapter in received Input X'7E'
000A	680	7,13	No CA level 1 request, or request from a CA other than the tested one, in Received Input X'6E' [1]
000B	680	8,13	No internal bus parity check or CCU interconnection card check in Received Input X'6D' [1]
000C	680	10,13	Level 1 request from adapter or IOC not reset in Received Input X'7E' [1]
000D	680	11,13	CA level 1 request not reset in Received Input X'6E' [1]
000E	680	12,13	Internal bus parity check or CCU interconnection card check not reset in Received Input X'6D'

[1] 'ERR BIT' = erroneous bits received

LA15 - COUNTER CHECK MECHANISM

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset
3. Output X'66' to set check checkers
4. Output X'6C' with pattern X'0088' with a bad parity
5. Check that a level 1 occurred
6. Input X'7E' to check level 1 request from adapter
7. Input X'6E' to check for CA level 1 from tested CA
8. Input X'6D' to check CCU interconnection card check
9. Output X'67' with reset level 1 checks
10. Input X'7E' to check no level 1 request from adapter or IOC
11. Input X'6E' to check no level 1 request from CA
12. Input X'6D' to check no CCU interconnection card check
13. Change error codes X'0001' through X'0007' to X'0008' through X'000E'

    - Rerun steps 3 through 12

14. CA diagnostic reset
15. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	5	No level 1 occurred after loading register X'6C' with a bad parity pattern
0002	680	6	No adapter level 1 request in received Input X'7E'
0003	680	7	No CA level 1 request, or request by a CA other than the tested one, in Received Input X'6E'
0004	680	8	No CCU interconnection card check in received Input X'6D'
0005	680	10	Always level 1 request from adapter or IOC after reset level 1 checks in Received Input X'7E' [1]
0006	680	11	CA level 1 request not reset in Received Input X'6E' [1]
0007	680	12	CCU interconnection card check not reset in Received Input X'6D' [1]
0008	680	5,13	No level 1 occurred after loading register X'6C' with a bad parity pattern
0009	680	6,13	No adapter level 1 request in received Input X'7E'
000A	680	7,13	No CA level 1 request, or request by a CA other than the tested one, in Received Input X'6E'
000B	680	8,13	No CCU interconnection card check in received Input X'6D'
000C	680	10,13	Always level 1 request from adapter or IOC after reset level 1 checks in Received Input X'7E' [1]
000D	680	11,13	CA level 1 request not reset in Received Input X'6E' [1]
000E	680	12,13	CCU interconnection card check not reset in Received Input X'6D' [1]

[1] 'ERR BIT' = erroneous bits of received pattern

LA16 - CHANNEL INTERFACE CARD CHECK

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'66' to set force error mode
4. On level 1, wait 2.5 seconds for driver check
5. Check that a level 1 occurred
6. Input X'7E' to check level 1 request from adapter
7. Input X'6E' to check for any level 1 from CA and for level 1 from tested CA
8. Input X'6D' to check internal bus parity check, channel interface card check, and driver receiver interface A
9. If TPS feature is defined in the CDF:
   - Input X'6F' to check that the TPS bit sent by the hardware
   - Input X'6D' to check internal bus parity check, channel interface card check, and driver receiver interface B.
10. If no TPS feature is defined in the CDF:
    - Input X'6D' to check no driver receiver interface B.
11. Reset level 1 checks and force error mode
12. Input X'7E' to check no level 1 request from adapter, nor IOC
13. Input X'6E' to check no level 1 from CA
14. Input X'6D' to check error bits correct reset
15. CA diagnostic reset
16. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	5	No level 1 occurred after set of force error mode
0002	680	6	No level 1 request from adapter in received Input X'7E'
0003	680	7	No CA level 1 request, or request from another CA than the tested one, in Received Input X'6E'
0004			Received Input X'6D' does not match expected pattern:
	682	8	- X'4802' (for interface A), RAC 682 if interface A error
	683	9	- X'4801' (for interface B), RAC 683 if interface B error ('ERR BIT' = erroneous bits of received pattern)
0005	680	12	Level 1 request from adapter or IOC not reset in Received Input X'7E' ('ERR BIT' = erroneous bits of received pattern)
0006	680	13	Level 1 request from CA not reset in Received Input X'6E' ('ERR BIT' = erroneous bits of received pattern)
0007		14	Error bits not reset in Received Input X'6D':
	681		- If internal bus parity check or channel interface and check not reset
	682		- If driver receiver check interface A not reset.
	683		- If driver receiver check interface B not reset. ('ERR BIT' = erroneous bits of received pattern)
0008	683	10	TPS not defined in CDF and driver receiver B in Received Input X'6F' ('ERR BIT' = erroneous bits of received pattern)
0009	683	9	TPS defined in CDF and no TPS bit in Received Input Input X'6F' [1] ('ERR BIT' = erroneous bits of received pattern)

[1] 'ERR BIT' = erroneous bits of received pattern

LA17 - NSC ACTIVE LATCH

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'67' to select CA (CA address, CA select, disable auto select)
4. Output X'62' to set NSC active (Output X'62' with pattern X'2000')
5. Input X'67' to check NSC active on
6. Output X'67' to reset CA (CA address, select bit, reset bit, disable auto select)
7. Output X'67' to reset CA (CA address, select bit, disable auto select)
8. Input X'67' to check NSC active off
9. Output X'62' to set NSC active (Output X'62' with pattern X'2000')
10. Input X'67' to check NSC active on
11. Output X'67' to reset NSC active (CA address, CA select, reset NSC active)
12. Input X'67' to check NSC active off
13. CA diagnostic reset
14. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	681	5	NSC active bit not set in Received Input X'67'
0002	681	8	NSC active bit not reset in Received Input X'67'
0003	681	10	NSC active bit not set in Received Input X'67'
0004	681	12	NSC active bit not reset in Received Input X'67'

LA19 - INITIATE SERVICE LATCH

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'66' to set check checkers
4. Output X'6C' with pattern X'0022' to load with bad parity
5. Software timer to wait for level 1
6. Check if level 1 occurred
7. Check CCU register X'7E' for an adapter level 1 request
8. Input X'6E' to check if level 1 is from tested CA
9. Input X'6D' to check internal bus parity error and CCU interconnection card check
10. Output X'67' to reset level 1 checks
11. Check CCU register X'7E' for no level 1 from adapter
12. Input X'6E' to check no level 1 from CA
13. Input X'6D' to check reset of internal bus parity check and CCU interconnection card check
14. Input X'60' to decrement cycle steal counter
15. Software timer to wait for level 1
16. Check if level 1 occurred
17. Check CCU register X'7E' for a level 1 request from adapter
18. Input X'6E' to check CA level 1 request
19. Input X'6D' to check internal bus parity check and CCU interconnection card check
20. Output X'67' to reset CA
21. Output X'67' to reselect CA
22. Input X'7E' to check no level 1 request from adapter or IOC
23. Input X'6E' to check no level 1 from CA
24. Input X'6D' to check no internal bus parity check or CCU interconnection card check
25. Output X'6C' to load cycle steal counter with X'0022'
26. Input X'60' to check no decrement of cycle steal counter
27. Input X'6C' to check cycle steal counter
28. CA diagnostic reset
29. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	6	No level 1 occurred after loading of register X'6C' with a bad parity pattern
0002	680	7	No level 1 from adapter in Received Input X'7E'
0003	680	8	No CA level 1 in Received Input X'6E'
0004	680	9	No internal bus parity check or CCU interconnection card check in Received Input X'6D'
0005	680	11	Always level 1 request from adapter or IOC in Received Input X'7E' ('ERR BIT' = erroneous bits under mask of Received Input X'7E')
0006	680	12	Always level 1 request in Received Input X'6E' ('ERR BIT' = erroneous bits under mask of Received Input X'6E')
0007	680	13	Always internal bus parity check or CCU inter connection card check in Received Input X'6D' ('ERR BIT' = erroneous bits under mask of Received Input X'6D')
0008	680	16	No level 1 occurred after decrement of bad register X'6C'
0009	680	17	No adapter level request in received Input X'7E'
000A	680	18	No CA level 1 request in Received Input X'6E'
000B	680	19	No internal bus parity check or CCU interconnection card card check in Received Input X'6D'
000C	680	22	Always level 1 request from adapter or IOC in Received Input X'7E'
000D	680	23	Always CA level 1 request in Received Input X'6E'
000E	680	24	Always internal bus parity check or CCU interconnection card check in Received Input X'6D'
000F	680	26	Cycle steal counter decrement in Received Input X'6C' ('ERR BIT' = erroneous bits of cycle steal counter)

LA1A - PRIORITY DATA/STATUS SERVICE

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'62' to set priority outbound service
4. Intput X'62' to check priority outbound service
5. Output X'62' with pattern zero
6. Input X'62' to check priority outbound service
7. Output X'62' with priority outbound service
8. Input X'62' to check priority outbound service
9. Output X'60' with any pattern
10. Input X'62' to check reset of priority outbound service
11. Output X'62' to set priority outbound service
12. Input X'62' to check priority outbound service
13. Output X'67' with reset system reset
14. Input X'62' to check reset of priority outbound service
15. Output X'62' to set priority outbound service
16. Input X'62' to check priority outbound service
17. CA diagnostic reset
18. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	4	No priority outbound service in Received Input X'62'
0002	680	6	No priority outbound service in Received Input X'62'
0003	680	8	No priority outbound service in Received Input X'62'
0004	680	10	Always priority outbound service after reset initial selection in Received Input X'62'
0005	680	12	No priority outbound 12 service in received Input X'62'
0006	680	14	Priority outbound service always present after reset system reset in Received Input X'62'
0007	680	16	No priority outbound service in Received Input X'62'

LA1B - OUTPUT SELECTION

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'67' (CA address, temporary selection, PRI, disable auto select)
4. Intput X'77' to check CA level 3 request
5. Input X'6F' to check data/status and CA address
6. Input X'62' to check program request interrupt
7. Output X'67' (CA address, permanent selection, disable auto select)
8. Input X'77' to check CA level 3 request
9. Input X'6F' to check data/status and CA address
10. Input X'62' to check program request interrupt
11. CA diagnostic reset
12. Disable redrives broadcast

ERC	RAC	Step	Error Description
0001	680	4	No CA level 3 request in Received Input X'77'
0002	680	5	No data/status or bad CA address in Received Input X'6F'
0003	680	6	No program request interrupt in Received Input X'62'
0004	680	8	Same as 0001 above
0005	680	9	Same as 0002 above
0006	680	10	Same as 0003 above

LA1D - OUTBOUND CYCLE STEAL (BUFFER A ONLY)

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'6C' to load cycle steal counter (with a count of 8)
4. Output X'3X' to load CS address register with an even address (X = CA address)
5. Clear buffer A in PIO mode by Outputs X'64', X'65', and X'66'
6. Output X'62' to start outbound operation in AIO mode
7. Software timer to wait for operation completion
8. Output X'67' to reset CA
9. Output X'67' to again select CA
10. Read buffer A in PIO mode by Inputs X'64' and X'65' and Output X'66'; expected patterns are X'0102', X'0304', X'0506', X'0708'
11. Run again steps 3 through 10, with cycle steal byte count = 8, CS address register loaded with an odd address; expected patterns are X'0203', X'0405', X'0607', X'0811'; change error codes from X'0001' through X'0004' to X'0011' through X'0014'
12. Run again steps 3 through 10, with cycle steal byte count = 1, CS address register loaded with an even address; expected patterns are X'0100', X'0000', X'0000', X'0000'; error codes are X'0021' through X'0024'
13. Run again steps 3 through 10, with cycle steal byte count = 2, address register on an even address; expected patterns are X'0102', X'0000', X'0000', X'0000'; error codes are X'0031' through X'0034'
14. Run again steps 3 through 10, with cycle steal byte count = 3, address register on an even address; expected patterns are X'0102', X'0300', X'0000', X'0000'; error codes are X'0041' through X'0044'
15. Run again steps 3 through 10, with cycle steal byte count = 4, address register on an even address; expected patterns are X'0102', X'0304', X'0000', X'0000'; error codes are X'0051' through X'0054'
16. Run again steps 3 through 10, with cycle steal byte count = 5, address register on an even address; expected patterns are X'0102', X'0304', X'0500', X'0000'; error codes are X'0061' through X'0064'
17. Run again steps 3 through 10, with cycle steal byte count = 6, address register on an even address; expected patterns are X'0102', X'0304', X'0506', X'0000'; error codes are X'0071' through X'0074'
18. Run again steps 3 through 10, with cycle steal byte count = 7, address register on an even address; expected patterns are X'0102', X'0304', X'0506', X'0700'; error codes are X'0081' through X'0084'
19. Run again steps 3 through 10, with cycle steal byte count = 8, address register on an even address; expected patterns are X'0102', X'0304', X'0506', X'0708'; error codes are X'0091' through X'0094'
20. CA diagnostic reset
21. Disable redrives broadcast

Note: 'ERR BIT' = erroneous received bits for all ERCs.

(LA1D)

ERC	RAC	Step	Error Description
0001	680	10	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0002	680	10	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0003	680	10	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0004	680	10	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
0011	680	10,11	Received buffer A bytes 0-1 do not match expected pattern X'0203' [1]
0012	680	10,11	Received buffer A bytes 2-3 do not match expected pattern X'0405' [1]
0013	680	10,11	Received buffer A bytes 4-5 do not match expected pattern X'0607' [1]
0014	680	10,11	Received buffer A bytes 6-7 do not match expected pattern X'0811' [1]
0021	680	10,12	Received buffer A bytes 0-1 do not match expected pattern X'0100' [1]
0022	680	10,12	Received buffer A bytes 2-3 do not match expected pattern X'0000' [1]
0023	680	10,12	Received buffer A bytes 4-5 do not match expected pattern X'0000' [1]
0024	680	10,12	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0031	680	10,13	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0032	680	10,13	Received buffer A bytes 2-3 do not match expected pattern X'0000' [1]
0033	680	10,13	Received buffer A bytes 4-5 do not match expected pattern X'0000' [1]
0034	680	10,13	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0041	680	10,14	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0042	680	10,14	Received buffer A bytes 2-3 do not match expected pattern X'0300' [1]
0043	680	10,14	Received buffer A bytes 4-5 do not match expected pattern X'0000' [1]
0044	680	10,14	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0051	680	10,15	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0052	680	10,15	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0053	680	10,15	Received buffer A bytes 4-5 do not match expected pattern X'0000' [1]
0054	680	10,15	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0061	680	10,16	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0062	680	10,16	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0063	680	10,16	Received buffer A bytes 4-5 do not match expected pattern X'0500' [1]
0064	680	10,16	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0071	680	10,17	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0072	680	10,17	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0073	680	10,17	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0074	680	10,17	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0081	680	10,18	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0082	680	10,18	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0083	680	10,18	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0084	680	10,18	Received buffer A bytes 6-7 do not match expected pattern X'0700' [1]
0091	680	10,19	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0092	680	10,19	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0093	680	10,19	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0094	680	10,19	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]

[1] 'ERR BIT' = erroneous received bits

LA1E - OUTBOUND CYCLE STEAL (BUFFERS A AND B)

1. Enable redrive specific (redrive address extracted from CA address)
2. CA diagnostic reset, and CA selection
3. Output X'6C' to load cycle steal counter (with a count of 16)
4. Output X'3X' to load CS address register with an even address (X = CA address)
5. Clear buffers A and B in PIO mode by Outputs X'64', X65', and X'66'
6. Output X'62' to start outbound operation in AIO mode
7. Software timer to wait for operation completion
8. Output X'67' to reset CA
9. Output X'67' to again select CA
10. Read buffers A and B in PIO mode by Inputs X'64' and X'65' [1] and Output X'66' [1]
11. Run again steps 3 through 10, with cycle steal byte count = 16, address register odd; expected patterns are X'0203', X'0405', X'0607', X'0811', X'1213', X'1415', X'1617', X'1819'; error codes are X'0011' through X'0018'
12. Run again steps 3 through 10, with cycle steal byte count = 1, address register even; expected patterns are X'0100', X'0000', X'0000', X'0000', X'0000', X'0000', X'0000', X'0000'; error codes are X'0021' through X'0028'
13. Run again steps 3 through 10, with cycle steal byte count = 2, address register even; expected patterns are X'0102', X'0000', X'0000', X'0000', X'0000', X'0000', X'0000', X'0000'; error codes are X'0031' through X'0038'
14. Run again steps 3 through 10, with cycle steal byte count = 3, address register even; expected patterns are X'0102', X'0300', X'0000', X'0000', X'0000', X'0000', X'0000', X'0000'; error codes are X'0041' through X'0048'
15. Run again steps 3 through 10, with cycle steal byte count = 4, address register even; expected patterns are X'0102', X'0304', X'0000', X'0000', X'0000', X'0000', X'0000', X'0000'; error codes are X'0051' through X'0058'
16. Run again steps 3 through 10, with cycle steal byte count = 5, address register even; expected patterns are X'0102', X'0304', X'0500', X'0000', X'0000', X'0000', X'0000', X'0000'; error codes are X'0061' through X'0068'
17. Run again steps 3 through 10, with cycle steal byte count = 6, address register even; expected patterns are X'0102', X'0304', X'0506', X'0000', X'0000', X'0000', X'0000', X'0000'; error codes are X'0071' through X'0078'
18. Run again steps 3 through 10, with cycle steal byte count = 7, address register even; expected patterns are X'0102', X'0304', X'0506', X'0700', X'0000', X'0000', X'0000', X'0000'; error codes are X'0081' through X'0088'
19. Run again steps 3 through 10, with cycle steal byte count = 8, address register even; expected patterns are X'0102', X'0304', X'0506', X'0708', X'0000', X'0000', X'0000', X'0000'; error codes are X'0091' through X'0098'
20. Run again steps 3 through 10, with cycle steal byte count = 9, address register even; expected patterns are X'0102', X'0304', X'0506', X'0708', X'1100', X'0000', X'0000', X'0000'; error codes are X'00A1' through X'00A8'
21. Run again steps 3 through 10, with cycle steal byte count = 10, address register even; expected patterns are X'0102', X'0304', X'0506', X'0708', X'1112', X'0000', X'0000', X'0000'; error codes are X'00B1' through X'00B8'
22. Run again steps 3 through 10, with cycle steal byte count = 11, address register even; expected patterns are X'0102', X'0304', X'0506', X'0708', X'1112', X'1300', X'0000', X'0000'; error codes are X'00C1' through X'00C8'
23. Run again steps 3 through 10, with cycle steal byte count = 12, address register even; expected patterns are X'0102', X'0304', X'0506', X'0708', X'1112', X'1314', X'0000', X'0000'; error codes are X'00D1' through X'00D8'
24. Run again steps 3 through 10, with cycle steal byte count = 13, address register even; expected patterns are X'0102', X'0304', X'0506', X'0708', X'1112', X'1314', X'1500', X'0000'; error codes are X'00E1' through X'00E8'
25. Run again steps 3 through 10, with cycle steal byte count = 14, address register even; expected patterns are X'0102', X'0304', X'0506', X'0708', X'1112', X'1314', X'1516', X'0000'; error codes are X'00E1' through X'00E8'

26. Run again steps 3 through 10, with cycle steal byte count = 15, address register even; expected patterns are X'0102', X'0304', X'0506', X'0708', X'1112', X'1314', X'1516', X'1700'; error codes are X'0101' through X'0108'
27. Run again steps 3 through 10, with cycle steal byte count = 16, address register even; expected patterns are X'0102', X'0304', X'0506', X'0708', X'1112', X'1314', X'1516', X'1718'; error codes are X'0111' through X'0118'
28. CA diagnostic reset
29. Disable redrives broadcast

(LA1E)

ERC	RAC	Step	Error Description
0001	680	10	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0002	680	10	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0003	680	10	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0004	680	10	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
0005	680	10	Received buffer B bytes 0-1 do not match expected pattern X'1112' [1]
0006	680	10	Received buffer B bytes 2-3 do not match expected pattern X'1314' [1]
0007	680	10	Received buffer B bytes 4-5 do not match expected pattern X'1516' [1]
0008	680	10	Received buffer B bytes 6-7 do not match expected pattern X'1718' [1]
0011	680	10,11	Received buffer A bytes 0-1 do not match expected pattern X'0203' [1]
0012	680	10,11	Received buffer A bytes 2-3 do not match expected pattern X'0405' [1]
0013	680	10,11	Received buffer A bytes 4-5 do not match expected pattern X'0607' [1]
0014	680	10,11	Received buffer A bytes 6-7 do not match expected pattern X'0811' [1]
0015	680	10,11	Received buffer B bytes 0-1 do not match expected pattern X'1213' [1]
0016	680	10,11	Received buffer B bytes 2-3 do not match expected pattern X'1415' [1]
0017	680	10,11	Received buffer B bytes 4-5 do not match expected pattern X'1617' [1]
0018	680	10,11	Received buffer B bytes 6-7 do not match expected pattern X'1819' [1]
0021	680	10,12	Received buffer A bytes 0-1 do not match expected pattern X'0100' [1]
0022	680	10,12	Received buffer A bytes 2-3 do not match expected pattern X'0000' [1]
0023	680	10,12	Received buffer A bytes 4-5 do not match expected pattern X'0000' [1]
0024	680	10,12	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0025	680	10,12	Received buffer B bytes 0-1 do not match expected pattern X'0000' [1]
0026	680	10,12	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
0027	680	10,12	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
0028	680	10,12	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
0031	680	10,13	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0032	680	10,13	Received buffer A bytes 2-3 do not match expected pattern X'0000' [1]
0033	680	10,13	Received buffer A bytes 4-5 do not match expected pattern X'0000' [1]
0034	680	10,13	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0035	680	10,13	Received buffer B bytes 0-1 do not match expected pattern X'0000' [1]
0036	680	10,13	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
0037	680	10,13	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
0038	680	10,13	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
0041	680	10,14	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0042	680	10,14	Received buffer A bytes 2-3 do not match expected pattern X'0300' [1]
0043	680	10,14	Received buffer A bytes 4-5 do not match expected pattern X'0000' [1]
0044	680	10,14	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0045	680	10,14	Received buffer B bytes 0-1 do not match expected pattern X'0000' [1]
0046	680	10,14	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
0047	680	10,14	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
0048	680	10,14	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
0051	680	10,15	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0052	680	10,15	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0053	680	10,15	Received buffer A bytes 4-5 do not match expected pattern X'0000' [1]
0054	680	10,15	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0055	680	10,15	Received buffer B bytes 0-1 do not match expected pattern X'0000' [1]
0056	680	10,15	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
0057	680	10,15	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
0058	680	10,15	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
0061	680	10,16	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0062	680	10,16	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]

[1] 'ERR BIT' = erroneous received bits

(LA1E, continued)

ERC	RAC	Step	Error Description
0063	680	10,16	Received buffer A bytes 4-5 do not match expected pattern X'0500' [1]
0064	680	10,16	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0065	680	10,16	Received buffer B bytes 0-1 do not match expected pattern X'0000' [1]
0066	680	10,16	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
0067	680	10,16	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
0068	680	10,16	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
0071	680	10,17	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0072	680	10,17	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0073	680	10,17	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0074	680	10,17	Received buffer A bytes 6-7 do not match expected pattern X'0000' [1]
0075	680	10,17	Received buffer B bytes 0-1 do not match expected pattern X'0000' [1]
0076	680	10,17	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
0077	680	10,17	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
0078	680	10,17	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
0081	680	10,18	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0082	680	10,18	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0083	680	10,18	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0084	680	10,18	Received buffer A bytes 6-7 do not match expected pattern X'0700' [1]
0085	680	10,18	Received buffer B bytes 0-1 do not match expected pattern X'0000' [1]
0086	680	10,18	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
0087	680	10,18	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
0088	680	10,18	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
0091	680	10,19	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0092	680	10,19	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0093	680	10,19	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0094	680	10,19	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
0095	680	10,19	Received buffer B bytes 0-1 do not match expected pattern X'0000' [1]
0096	680	10,19	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
0097	680	10,19	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
0098	680	10,19	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
00A1	680	10,20	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
00A2	680	10,20	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
00A3	680	10,20	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
00A4	680	10,20	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
00A5	680	10,20	Received buffer B bytes 0-1 do not match expected pattern X'1100' [1]
00A6	680	10,20	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
00A7	680	10,20	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
00A8	680	10,20	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
00B1	680	10,21	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
00B2	680	10,21	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
00B3	680	10,21	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
00B4	680	10,21	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
00B5	680	10,21	Received buffer B bytes 0-1 do not match expected pattern X'1112' [1]
00B6	680	10,21	Received buffer B bytes 2-3 do not match expected pattern X'0000' [1]
00B7	680	10,21	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
00B8	680	10,21	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
00C1	680	10,22	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
00C2	680	10,22	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
00C3	680	10,22	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
00C4	680	10,22	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]

[1] 'ERR BIT' = erroneous received bits.

(LA1E, continued)

ERC	RAC	Step	Error Description
00C5	680	10,22	Received buffer B bytes 0-1 do not match expected pattern X'1112' [1]
00C6	680	10,22	Received buffer B bytes 2-3 do not match expected pattern X'1300' [1]
00C7	680	10,22	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
00C8	680	10,22	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
00D1	680	10,23	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
00D2	680	10,23	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
00D3	680	10,23	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
00D4	680	10,23	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
00D5	680	10,23	Received buffer B bytes 0-1 do not match expected pattern X'1112' [1]
00D6	680	10,23	Received buffer B bytes 2-3 do not match expected pattern X'1314' [1]
00D7	680	10,23	Received buffer B bytes 4-5 do not match expected pattern X'0000' [1]
00D8	680	10,23	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
00E1	680	10,24	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
00E2	680	10,24	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
00E3	680	10,24	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
00E4	680	10,24	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
00E5	680	10,24	Received buffer B bytes 0-1 do not match expected pattern X'1112' [1]
00E6	680	10,24	Received buffer B bytes 2-3 do not match expected pattern X'1314' [1]
00E7	680	10,24	Received buffer B bytes 4-5 do not match expected pattern X'1500' [1]
00E8	680	10,24	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
00F1	680	10,25	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
00F2	680	10,25	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
00F3	680	10,25	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
00F4	680	10,25	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
00F5	680	10,25	Received buffer B bytes 0-1 do not match expected pattern X'1112' [1]
00F6	680	10,25	Received buffer B bytes 2-3 do not match expected pattern X'1314' [1]
00F7	680	10,25	Received buffer B bytes 4-5 do not match expected pattern X'1516' [1]
00F8	680	10,25	Received buffer B bytes 6-7 do not match expected pattern X'0000' [1]
0101	680	10,26	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0102	680	10,26	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0103	680	10,26	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0104	680	10,26	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
0105	680	10,26	Received buffer B bytes 0-1 do not match expected pattern X'1112' [1]
0106	680	10,26	Received buffer B bytes 2-3 do not match expected pattern X'1314' [1]
0107	680	10,26	Received buffer B bytes 4-5 do not match expected pattern X'1516' [1]
0108	680	10,26	Received buffer B bytes 6-7 do not match expected pattern X'1700' [1]
0111	680	10,27	Received buffer A bytes 0-1 do not match expected pattern X'0102' [1]
0112	680	10,27	Received buffer A bytes 2-3 do not match expected pattern X'0304' [1]
0113	680	10,27	Received buffer A bytes 4-5 do not match expected pattern X'0506' [1]
0114	680	10,27	Received buffer A bytes 6-7 do not match expected pattern X'0708' [1]
0115	680	10,27	Received buffer B bytes 0-1 do not match expected pattern X'1112' [1]
0116	680	10,27	Received buffer B bytes 2-3 do not match expected pattern X'1314' [1]
0117	680	10,27	Received buffer B bytes 4-5 do not match expected pattern X'1516' [1]
0118	680	10,27	Received buffer B bytes 6-7 do not match expected pattern X'1718' [1]

[1] 'ERR BIT' = erroneous received bits for all ERCs.

LA20 - AUTO SELECTION TEST

Because routine LA20 works interactively on several channel adapters, before running this routine, ensure that all other routines run without errors on all implemented channel adapters.

LA20 will only be available when the CA IFT is run on the last CA. When CA IFT is run on another CA, LA20 is not really effective, it is only a dummy run without tests.

1. Reset all implemented channel adapters
2. Prepare one channel adapter (named CA A)
3. Prepare another channel adapter (named CA B)
4. Send a permanent selection on CA A with program request interrupt set
5. Send a permanent selection on CA B with program request interrupt set
6. Enable auto select mechanism
7. Start a software timer to wait for potential levels 1 or 3
8. If a level 1 occurred, set current test in error and go to 12
9. If no level 3 present, set current test in error and go to step 12
10. Check for data/status level 3 request from CA A; if none, set current test in error and go to step 12
11. Input X'62' to check for program request interrupt; if none, set current test in error and go to step 12
12. Reset channel adapters A and B, and log test result
13. Prepare new CA B
14. Check if all CA Bs have been tested; if not, return to step 4 with a new CA B
15. Prepare new CA A
16. Check if all CA As have been tested; if not, return to step 4 with a new CA A, and reinitiate CA B
17. Result analysis

ERC	RAC	Error Description
0001	686	Complete auto selection test bad, or unable to analyze result
0002	684	One CA is suspected faulty
0003	685	Two CAs are suspected faulty (Second CA address is given in ADDIT INFO field)

LA LEVEL 1 - LEVEL ONE INTERRUPT HANDLER

1. Input CCU register X'7E'
2. Input CCU register X'76'
3. If IFT never entered, go to step 25
4. If last command was enable redrive, go to step 28
5. If several CAs under test, go to step 30
6. If AIO bit present in Input X'76', go to step 33
7. If no adapter level 1, nor level 1 summary, go to step 25
8. Output X'67' to be sure that CA under test is selected
9. Output X'76' to reset IOC errors
10. Input X'6E' to send CA level 1 register
11. Input X'76' to check if new error; if yes, go to step 25
12. If force error bit is on, send 3-second timer to wait for driver check line
13. Input X'6D' to get selected CA level 1 register
14. If no level 1 from tested CA, go to step 25
15. If level 1 not expected, go to step 36
16. If force error or check checkers switches off, go to step 18
17. Output X'66' with pattern X'0000' to clear check checkers and force error mode
18. If CA reset requested, output 67 to reset CA, then Output X'67' to select again CA under test
19. If only reset level 1 checks requested, Output X'67' to reset level 1 checks
20. Input X'6E' to check CA level 1 register
21. If always CA level 1 request, go to step 25
22. Input CCU register X'7E' after reset
23. Input X'6D' to check that level 1 is reset
24. Exit from level 1
25. Output X'77' to do CCU miscellaneous reset
26. Disable redrives broadcast
27. Send abend X'690'; disruptive sequence, only IPL allowed
28. Disable redrives broadcast
29. Send abend X'6A0'; disruptive sequence, only IPL allowed
30. If auto select switch off, go to step 32
31. Reset two CAs under test; then go to step 24
32. Send abend X'690'; disruptive sequence, only IPL allowed
33. Reset IOC errors by Output X'76'
34. Reset CA by Output X'67' with reset bit
35. Send error message (ERC X'0000'); all DCF options are available to continue, loop, abort, etc....
36. Decode Received Input X'6D'; send ERC X'0000' and update RAC with X'6D' value; all DCF options are available to continue, loop, abort, etc....

ERC/ Abend	RAC	Error Description	
abend 690		Unexpected level 1 interrupt - if IFT never entered - if no level 1 from adapter - if no level 1 from IOC - if unable to read CA level 1 register - if no level 1 from CA under test - if unable to reset expected level 1	ADDIT INFO field is filled - X'7E' - X'76' - last command sent - data only for a write command - CA number
abend 6A0		Unable to enable redrive specific	ADDIT INFO field gives message: "enable redrive failed"
0000	691	Unexpected level 1 during cycle steal operation ('ERR BIT' = Input X'76' contents)	
0000		Unexpected level 1 ('ERR BIT' = Input X'6D' contents)	
	680	- CCIN, CHIN	
	681	- CHIN, CCIN	
	682	- CADR A, CHIN, CCIN	
	683	- CADR B, CHIN, CCIN	

MA - OLT RESPONDER

The MA OLT Responder section is loaded into the 3725 CCU from the Service Diskette when function 6 is requested from the Diagnostic general menu.

INITIALIZATION SETUP

The initialization does the following at level 4:

- Unmask level 1 and level 3 interrupt
- Enable the redrives
- Prompt CE to request the setting of ENA/DIS switch to have ESC operational and to release the channel if it is hung
- Select the CA under test
- Request PRI level 3 when requested by the Host
- Wait for SHUT DOWN

RESPONDER LEVEL 3 PROCESSING

Level 3 of the Responder is dispatched when communication has to be established with the host.

The causes of level 3 interrupt are:

- Initial selection (I/S)
- Data status (D/S)
- 100-ms timer

1. I/S processing: The I/S is originated by the following causes found from In X'60' of the CA

   - Status cleared
   - System reset
   - Command decode
   - Initial status stacked
   - Selective reset
   - Interface stop

2. D/S processing: The D/S is originated by the following causes found from In X'62' of the CA

   - Suppress out monitor
   - Interface stop
   - Outbound data transfer sequence
   - Inbound data transfer sequence
   - Status transfer sequence
   - Program level 3 interrupt request
   - Status is stacked
   - ESC final status transfer

3. 100-ms timer: The 100-ms timer is used by the PRI routine (when requested) to keep the Responder in loop into level 3 to test the busy condition of the CA

I/S COMMAND DECODE

The following commands (1st byte of Host CCW) cause level 3 interrupt:

- Write Break (X'09') WTBRKCMD
- Write IPL (X'05') WRAPCMD
- NOP (X'03') NOOPCMD
- Read (X'02') READCMD
- Write (X'01') WRITECMD
- Sense (X'04') SENSECMD
- Sense ID (X'E4') SIOCMD

In addition, the TIO instruction is also processed as a command. Any other command is rejected with CE, DE, UC as final status.

D/S PROCESSING

The D/S interrupt is processed according to the analysis done when In X'62' is performed.

- Suppress out monitor: SUPROUT

  This routine is dispatched when the status has been stacked and status has to be transferred. It resets D/S and exits.

- Interface stop

  This routine is dispatched when the CCW count at the host side has gone to zero or when a HIO has been issued. The routine empties the CA buffer and prepares the status CE, DE to be transferred.

- Outbound data transfer sequence: OUTDATA

  This routine is dispatched when a data transfer is in progress from the CA to the Host. The routine sends back to the host the exact number of bytes received by the previous transfer.

  The transmission from CCU storage to the CA buffer is done either in PIO or in cycle steal mode, depending on the request coming from the host.

- Inbound data transfer sequence: INDATA

  This routine is dispatched when a data transfer from the host to the CA is in progress. The routine moves the data from the CA buffers (In X'64', In X'65') into the Read area.

  The count of received data is bumped by 4 if four bytes were received, and sets up a new request for four bytes.

- Status transfer sequence

  This routine is dispatched when the host received and accepted the status.

  It determines if the Write is for a shutdown; if so, it resets the shut-down switch to be used by level 4.

  It resets D/S and exits.

- Program level 3 interrupt request: PGMJRPT

  This routine is dispatched when level 4 has requested a PRI.

  The routine keeps looping for 3 seconds into level 3

- Status is stacked: ENDSTACK

  This routine is dispatched when the host has not yet accepted the final status which is stacked.

  The routine requests the CA to monitor the suppress out via an Out X'67' and exits.

- ESC final status transfer: ESCSTAT

  This routine is dispatched when final status for an ESC operation has been received and accepted by the host.

  The routine sets ESC command free and resets D/S to exit.

LEVEL 1 INTERRUPT HANDLER

Level 1 interrupt handler determines the origin of an interrupt:

- If it is an IOC Level 1, then:

  RAC is 663
  ERC is X'7999'
  ERR bits gives the In X'76' (first 2 digits)

- If it is an Adapter Level 1, then:

  RAC is 663
  ERC is X'7998'
  ERR bits gives the CA In X'6D' (all bits)

- If it is none of above interrupts, then the message L1 RELOAD is displayed.

CHAPTER 5. TSS DIAGNOSTICS

Table of Contents for Chapter 5

CHAPTER 5. TSS DIAGNOSTICS

The TSS diagnostic group is divided into three IFTs that test:

- FES (IFT P)
- LIC (IFT Q)
- ICC (IFT R)

This group tests the FES card, the LIC cards, and ICC cards that are present on the scanner.

Note: The CSP and CSM cards are tested during the scanner IML using the microcode taken from its ROS as part of a scanner IML or running the IOC bus IFT.

The TSS group runs under the control of the DCM in the MOSS. The command processor and the IFTs are loaded in the scanner to be tested.

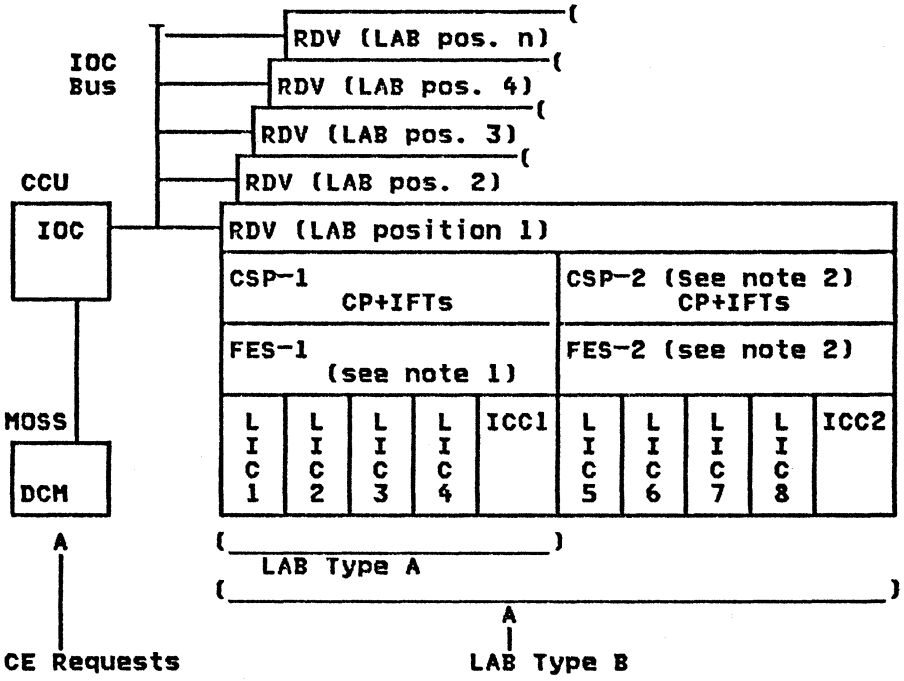


Figure 5-1. TSS Diagnostics

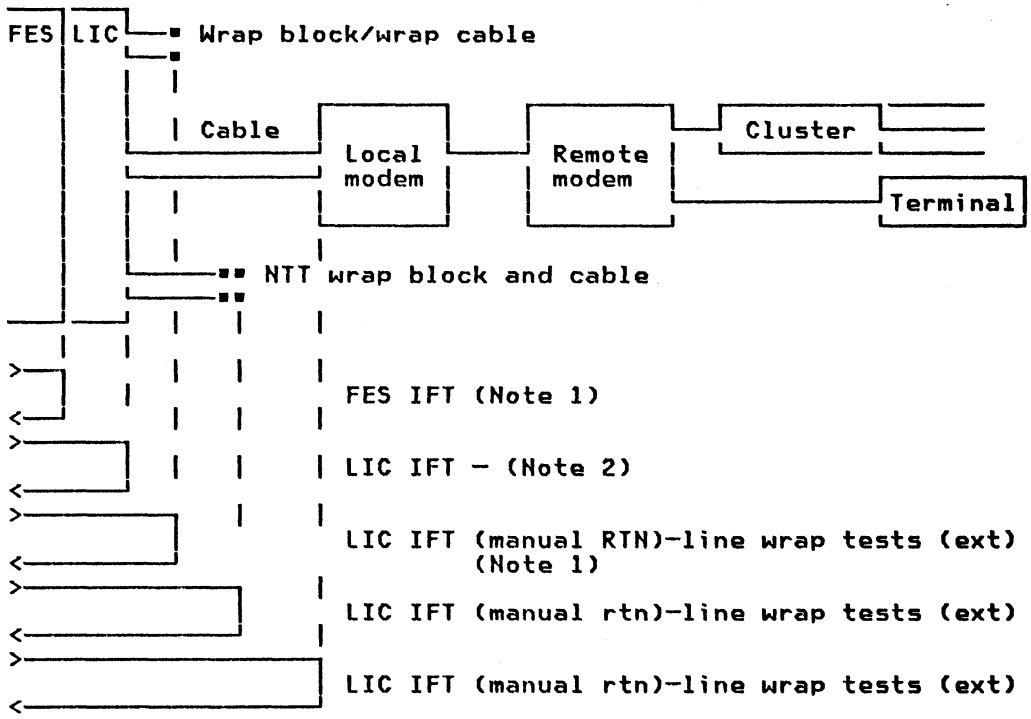
Notes:

1. The interface between the FES and the LIC can be disabled by the diagnostic program to prevent LIC interference on the FES card.
2. This block diagram applies only to a LAB type B. In a LAB type A, there is neither CSP-2 nor FES-2.

LINE TESTING POSSIBILITIES

The following figure shows the different wrap test possibilities controlled from MOSS on the communication link, in particular, the progression of testing procedures from the TSS to the terminal.

CONTROLLED FROM THE MOSS



Notes:

1. A line position can be plugged with a line cable, or be without a line cable, or can be plugged with a wrap block (LIC type 1, 2, 4A, or 4B), or with a wrap cable (LIC type 3). The CDF for each line must be updated accordingly.

   When the TSS IFTs are run, the hardware for a selected line is:

   a. Tested up to the LIC drivers if the line cable is present.

   b. Tested up to the LIC and ICC (if present) card level for a line without cable.

   c. Fully tested if a wrap block or a wrap cable is present on the selected line. Plugging a wrap block or wrap cable automatically selects the manual intervention section QC.

   d. In order to fully test the LIC3 card, it is necessary to reverse the LIC3 wrap cable after a first test pass, then run the test again.

2. For wrap during normal operation, see 'LIC Driver Check', on page 13-261 of the MIM-1.

REQUIREMENTS

Before running the TSS diagnostic group you must ensure that the CCU, IOC bus, and CA diagnostic groups work properly. If not, the results given by the TSS diagnostic group may be of no value, or misleading.

SELECTION

For running offline diagnostics, see Chapter 1.

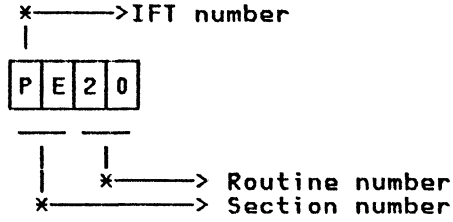
DIAG==>_

5	TSS group selected
X	Specific IFT X in this group
XY	Specific section y in IFT X
XYZZ	Specific routine ZZ in section XY

For specific section and routine selection, see routine lists on following pages, and in Chapter 1.

Move the cursor from its initial position (DIAG==>) to the next after each parameter is entered. To skip a parameter entry, press the —> key.

To interpret correctly the results of a selected section or routine, make sure the preceding IFTs, sections, and routines in the group are running without error. The routine identification contains the IFT number, the section number, and the routine number as follows:



ADP#==>_ Enter the scanner number in the range 1 to 16, except 2 or 4, which are not valid.

If no scanner is selected, the diagnostic will run on all scanners defined in the CDF.

LINE==> Enter the line address in the range 0 to 31.

When no line is selected in the request, routines PBxx, PCxx, and PExx automatically select the first line in the scanner (usually line address 00). This address is displayed in any message. However, you may select any line address, and it will be used to exercise the FES card. This address will be displayed in the messages.

OPT==> N For option display and description, see Chapter 1.

NUMBER OF RUNS PER REQUEST

The following table indicates how many times a section is run according to the selection request.

Select ADP#	Select LINE#	Number of Runs per Request
No	No	PA and PF: once per scanner PB to PE: once on the first installed line of each scanner QA: once per LIC QB: once per line RA: once per ICC RB: once per line RC: once per ICC
Yes	No	As above for the selected scanner
Yes	Yes	PA and PF: once PB to PE: once on the selected line QA: once on the LIC of the selected line QB: once on the selected line RA: once on the ICC of the selected line RB: once on the selected lines RC: once on the ICC of the selected line

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK

D:DIAGNOSTICS                                  SP:CCU STOP    Q:DATE/TIME
                                 U:UTILITY PGM ST:CCU START   T:TERMINATE
E:ERROR LOG                                    RS:CCU RESET

DIAG  |ADP# |LINE |
      |     |     |
1 ALL |     |     |
2 CCU |     |     |
3 IOCB|     |     |
4 CA  |1-> 6|     |
5 TSS |1->16|0->31|
6 OLT |1-> 6|     |
7 TRSS|6->16|1->4 |
       AND    |                          DIAG - RUN INIT
      |     |     |
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> PA    ADP#==> 5   LINE==> 0   OPT==> N
===>
```

On the above screen, section PA will run on line address 0 of the scanner number 5.

Press SEND to execute the request.

Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

MANUAL INTERVENTION ROUTINES

QCXX: Modem in register and data receiver.

QDXX: Modem in wrap test for Japan only.

TSS DIAGNOSTIC GROUP RUNNING TIME

The times below are related to a TSS diagnostic group request (DIAG=5) for **one specific scanner** (ADP# = x).

- If the scanner has 1 or 2 ICCs installed, the running time is:

  2 min 13 sec + (4 sec *N) + (13 sec *M) + (5 sec *L)

- If the scanner has no ICC installed, the running time is:

  1 min 58 sec + (4 sec *N) + (3 sec *L)

Where

N is the number of installed LICs
M is the number of installed ICCs
L is the number of installed Lines

FES IFT

The different sections of the FES IFT check the following:

- Asynchronous access
- Scanner base layer
- Modem management, start-stop, scanning
- BSC in transmit and receive

RUNNING TIME

The running time shown in the following table applies to a single scanner only.

Run Init	36 sec (see note 1)
PA	17 sec
PB	14 sec
PC	11 sec
PD	10 sec
PE	11 sec
PF	(see note 2)

Notes:

1. Run init = 1 min 25 sec when the CCU was not initialized by a previous diagnostic request.
2. 9 sec plus 1.5 times the total number of installed LICs (time in seconds).

PA ROUTINES

The PA section uses every LIC card and ICC card installed and defined in the 3725 CDF. The CDF must show at least one line installed on a LIC card for the LIC card to be used.

The PA section tests only the connection between the FES, the LIC, and the ICC cards.

The complete LIC card testing is performed by IFT(Q).

Messages

None

LIC IFT

Ports of a LIC that do not show a line installed in the CDF are not tested.

The QAxx routines run once on each LIC. Without line selection, the QBxx routines are repeated on each line attached on the LIC and defined in the CDF. The QCxx routines are run once on the specified line. QDxx routines are specific to the Nippon Telegraph Telephone (NTT) administration.

Details of how and when to use these manual intervention routines can be found in the MIM Part 2, Chapter 3.

RUNNING TIME

The following running times apply to a single scanner only.

QA	5 sec plus 2.5 times the total number of installed LICs (time in seconds)
QB	5 sec plus 3 times the total number of installed lines (time in seconds)

MANUAL INTERVENTION ROUTINES

QCxx and QDxx are manual intervention routines for Worldwide and Japan, respectively.

Worldwide (Wrap Test at Tailgate)

Routines QC01 through QC07, when selected, require you to plug a wrap block or a wrap cable instead of the modem connector on the 3725 or 3726 tailgates. They run automatically when the TSS group or the IFT Q is selected, provided that the wrap block (LIC type 1, 2, or 4) or the wrap cable (LIC type 3) is plugged on the selected line and the CDF is properly updated.

If it is not:

1. Place the wrap block or wrap cable on the selected line.
2. Update the cable identification "I" and clock information "C" if no clock is defined (clock value must be non-zero), using the CDF display/update scanner (see page 2-403 in MIM Part 1).
3. Select QC on the selected scanner and line.

Japan Only

Routines QD01 through QD05 are reserved for the Nippon Telegraph Telephone (NTT) administration. They check the data wrap regardless of the LIC type. They also check the modem control leads depending on the LIC type (modem-in wrap).

They must be selected; they do not run automatically like the QCxx routines.

QD01: NTT On/Off Driver Test: This routine sets permanently on or off all the used line drivers of a LIC card to allow measurements by the NTT service personnel.

The routine must be specifically selected together with the selected scanner and line, as shown in the following example:

```
DIAG==> QD01   ADP#===> 5   LINE==> 2   OPT==> N
```

When the message:

```
LINE DRIVER STATE: ON=F1, OFF=F2, EXIT=F9
```

is displayed, enter:

- RF1 to set drivers at high voltage level
- RF2 to set drivers at low voltage level
- RF9 to exit the routine

If you enter RF1 or RF2, the following message is displayed:

```
CHECK IF DRIVERS ARE AS REQUESTED.  PRESS SEND TO CONTINUE
```

At this step, the NTT personnel may check the driver voltage. To change the option, press SEND.

QD02: NTT Data Wrap Test: This routine checks the data wrap path (transmit to receive) regardless of the LIC type. The Test/Operate switch on the cable connector or on the DCE must be set as follows:

- LIC type 1: Set the connector Test/Operate switch to TEST.
- LIC type 3: Set the DCE Test/Operate switch to T1.
- LIC type 4x: Set the DCE Test/Operate switch to T1.

QD03, QD04, QD05: NTT Modem-In Wrap Test: These routines check the modem control leads according to the LIC type. Use the TEST/OPERATE switch or the wrap block as follows:

- LIC type 1 (V24): Set the connector TEST/OPERATE switch to TEST.
- LIC type 1 (V25): Plug the wrap block at the cable end.
- LIC type 3: Set the DCE Test/Operate switch to T1.
- LIC type 4x: Set the DCE Test/Operate switch to T1.

Messages

None

ICC IFT

The RAxx and RCxx routines run twice (ICC-1 and ICC-2). Without line selection, the RBxx routines are repeated on each line attached on the LIC and defined in the 3725/3726 CDF. The RBxx routines are run once if a line is specified.

RUNNING TIME

The following running times apply to a single scanner only.

RA	5 sec plus 3 times the total number of installed ICCs (time in seconds)
RB	5 sec plus twice the total number of installed lines (time in seconds).
RC	5 sec plus 10 times the total number of installed ICCs (time in seconds)

Messages

None

ROUTINES DESCRIPTION

CSP000 - CSP-ROS START-UP INITIALIZATION

This routine is given control when a reset pulse is received by the hardware CSP, which latches the reset into external register XR04. Two types of reset are processed:

1. POR (power-on reset) XR04 bit 2.
2. Program reset XR04 bit 1 (PIO sent by the CCU).

The routine tests the conditions set up by the POR (condition code and LSPR). If it is a program reset, the routine checks if the IOC bus IFT originated the reset; if so the responder is given control otherwise control is given to the scanner control code (CHHCRORT).

STEP - FUNCTION

1. After a reset, the microcode of the CSP is always given control at address X'0'.

2. Find the origin of the reset from CSP external register XR04 (bits 2 and 1).

   Error: Error found in testing the conditions set up by the POR (FIC: 05). (**FIC** is the FRU isolation code (see MIM Part 2).)

3. If POR, check that the condition code register is all X'0' and the LSPR is X'0'.

4. If it is a program reset then determine the origin of the reset.

5. If it is from IOC bus IFT then give control to the IOC bus responder.

6. Else give control to the SCC at address CHHCRORT.

CSP012 - TEST CSP BRANCH MICROINSTRUCTIONS

This routine tests the following CSP microinstruction:

- B Unconditional Branch
- BAL Branch and Link
- BALR Branch and Link Register
- BON Branch on Bit 'on' for local store and external register
- BOFF Branch on Bit 'off' for local store
- BC Branch on Condition

STEP - FUNCTION:

1. Test Unconditional Branch B instruction code reporting error following the branch.

   Error: Branch not taken (FIC: 05).

2. Test the Branch and Link instruction BAL. When BAL executes, the link address is saved in LS04-05.

   Error: The link address is not set in LS04-5 (FIC: 05).

3. Test Branch and Link The address saved by BAL is incremented by one to point to another BALR and execute.

   Error: BALR test fails (FIC: 05).

4. Test Branch on Bit On (Bon). The local store register LS0 is initialized to X'FF' and each local store position (8 bits) is tested by BON. Each BON branch address points to the next BON and so on.

   Error: BON test fails (FIC: 05).

5. Test Branch on Bit Off (BOFF) (negative test). LS0 still contains X'FF'. Each bit position of LS0 is tested and the branch address points to error reporting.

   Error: Branch occurred while the LS0 bits are on (FIC: 05).

6. Test Branch on Bit On (Bon) (negative test). The LS0 is set to X'0' and each bit position of LS0 is tested using the BON microinstruction. The Branch address points to error reporting.

   Error: Branch occurred while the LS0 bits are all off (FIC: 05).

7. Test Branch on Bit Off (BFF) (positive test). LS0 still contains X'0' and each bit position of LS0 is tested. The branch address points to the next BOFF and so on for each bit.

   Error: LS0 has all bits off, and BOFF fails to branch (FIC: 05).

8. Test Branch on Bit On for CSP external register. The XR used is the LSPR which is in the CSP microprocessor

9. The LSPR is loaded with X'FF'. Each bit position of LSPR (except bit 3) is tested one after the other starting at bit 0. The branch address points to the next BON and so on.

   Error: XR31 (LSPR) has all bits on and BON fails to branch (FIC: 05).

10. Test Branch if Any Bit On (BANY). The LSPR is loaded with the following values:

    X'FF', X'01', X'02', X'04', X'08', X'10', X'20', X'80'.

    For each pattern BANY is issued; the branch address points to the next BANY. The LSPR is then set to X'0' and the BANY branch address points to error reporting.

Error: BANY failed to branch (FIC: 05).

11. Test BON on XR using LSPR (negative test). LSPR being set to X'0' from previous test, each bit position of LSPR is tested using BON whose branch address points to error reporting.

    Error: Branch occurred while XR31 is all X'0' (FIC: 05).

    Test Branch on Condition. The condition code is a 3-bit register as follows:

    Error: Branch on Condition failed (FIC: 05).

    - EQUAL/ALL
    - ZERO/NONE
    - Carry

    The following ALU codes are used to determine the condition:

    - ZERO/NONE
    - NOT ZERO/NOT NONE
    - CARRY
    - NOT CARRY
    - EQUAL/ALL
    - NOT EQUAL/NOT ALL

12. The local store register LS0 is set to X'0', then test BZ (Branch if Zero), BNZ (Branch if not zero), and BNE (Branch if not equal).

    LS0 is set to 01 and tests:

    BZ, BNZ, and BC (Branch if Carry).

    The CC is set using a compare and test BE (Branch if equal) and LS01 is incremented by 1 until the carry condition is set, then test BNC (Branch if Not Carry) and BC (Branch if Carry).

FIC is the FRU isolation code (see MIM Part 2).

CSP026 - TEST LOAD REGISTER IMMEDIATE (LRI) MICROINSTRUCTION

This routine tests the Load Register Immediate (LRI) instruction via a set of patterns loaded into the local store registers of page X'0'. Both primary and secondary pages are identical in the LSPR. The test is done in such a way that to each register corresponds a value having a meaning as a bit position there a set of pattern having a meaning as number of bits loaded.

STEP - FUNCTION:

1. The LSPR is all X'0' when the routine is started which means that primary and secondary pages are the same. Set up LS0 through LS7 with the following values:
   X'01', X'02', X'04', X'08', X'10', X'20', X'40', X'80'
   and check them.

   Error: Data pattern do not compare (FIC: 05).

2. Set up LS8 to LS15 with the following values:
   X'FF', X'EF', X'EE', X'CE', X'CC', X'8C', X'88', X'00'
   and check them.

   Error: Data pattern do not compare (FIC: 05).

FIC is the FRU isolation code (see MIM Part 2).

TEST REGISTER IMMEDIATE (RI) MICROINSTRUCTIONS

This routine tests the following set of microinstructions (RI format):

- ARI ADD Immediate
- ACRI ADD with Carry Immediate
- ORI OR Immediate
- XRI XOR Immediate
- NRI AND Immediate
- CRI COMP Immediate
- TRI Test Under Mask Immediate
- SRL Shift Right One Position

The test is done in the following order:

ARI, ACRI, ORI, XRI, NRI, SRL, CRI, TRI.

A set of patterns are added to LS0 and checking is done using the CC set up by ARI and ACRI.

Data patterns are ORed and ANDed to set up the CC used to test ORI, XRI and NRI. LS0 is set to X'01' and SRL is issued. The CC should be X'0'. The following set of patterns are used to test CRI: X'00', X'01', X'FF'

TRI is tested using the following patterns:

X'EE' (MSK = X'AA', X'00')
X'A6' (MSK = X'5A')
X'F0' (MSK = X'1A')

Error: RI instruction(s) failed (FIC: 05). (**FIC** is the FRU isolation code)

TEST EXTERNAL REGISTER IMMEDIATE (XI) MICROINSTRUCTIONS

This routine tests the following set of microinstructions (XI format):

- LX Load Register
- CX Compare Register
- AXI Add Register Left (4 bits) and Right (4 bits)
- LXI Load Register Left (4 bits) and Right (4 bits)
- OXI OR Immediate Left (4 bits) and Right (4 bits)
- XXI XOR Immediate Left (4 bits) and Right (4 bits)
- NXI AND Immediate Left (4 bits) and Right (4 bits)
- TXI Test Under Mask Immediate Left (4 bits) and Right (4 bits)

FUNCTION: To test the XI type of microinstruction the LSPR (XR31) residing in the CSP is used.: Except for LX and CX instructions, all handle 4 bits of addressed XR. The handling of the 4 bits can be specified as a modifier in the instruction.

If 'left' is specified, then only bits 0 to 3 of the XR are involved. If 'right' is specified, then only bits 4 through 7 of the XR are involved.

The CC is the means used to check each instruction. All XI instructions, when they execute, set up the CC. The Branch on Condition following the execution of the instruction determines if the instruction executes correctly.

STEP:

1. LX and CX: local store registers LS0 and LS1 are initialized to do the test.

   The following Branch on Condition instructions are used to test the result: BZ, BNE, BE

   Error: The LX or CX instruction failed to set CC (FIC: 05).

2. AXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BNC, BZ, BC, BNE.

3. LXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BZ, BNE.

4. OXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BZ, BNE.

5. XXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BZ, BNE.

6. NXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BZ, BNE.

7. TXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNO, BN, BNN, BO.

Error: The XI microinstruction (AXI, LXI, OXI, XXI, NXI, or TXI) failed to set CC (FIC: 05).

Note: For each Branch on Condition used, the Branch address points to error reporting.

TEST REGISTER TO REGISTER (RR) MICROINSTRUCTIONS

This routine tests the following set of microinstructions (RR format):

- AR ADD
- ACR ADD with Carry
- OR OR Logical
- XR XOR
- NR AND
- CR Compare
- TMR Test Under Mask
- LR Load Register
- LHR Load Register Double

To test the RR type of microinstruction, 2 LS registers must be initialized. They are loaded using LRI. LS0 and LS1 (page X'0') are used for all RR tests (except LHR). The CC is the mean used to check each instruction. All RR instructions when they execute set up the CC and the Branch on Condition shows the way the instruction has executed.

STEP - FUNCTION:

1. AR: the following CCs are tested: BZ, BC, BNZ, BNC.

2. ACR: the following CCs are tested: BC, NZ, BNE, BNC,and BNZ.

3. OR: the following CCs are tested BNZ, BZ, BC.

4. XR: the following CCs are tested: BNZ, BZ, and BNE

5. NR: the following CCs are tested: BNZ, BZ, and BNE

6. CR: the following CCs are tested: BNE and BE.

7. TMR: the following CCs are tested: BNO, BN, BNN, BO.

   The value is loaded into LS0 and the mask applied is loaded into LS1.

   Error: LR failed to set CC (FIC: 05).

8. LR: the following CCs are tested: BZ, BNE, and BNZ.

9. LHR: each pair of local store registers LS0, LS2, LS4, compare. LS6 is loaded with the same value and CRI is used to check for correct loading.

   Error: LHR failed during the compare (FIC: 05).

Error: The RR microinstruction (AR, ACR, OR, XR, NR, CR, TMR, LR) failed to set CC (FIC: 05).

Note: For each Branch on Condition used, the Branch address points to error reporting.

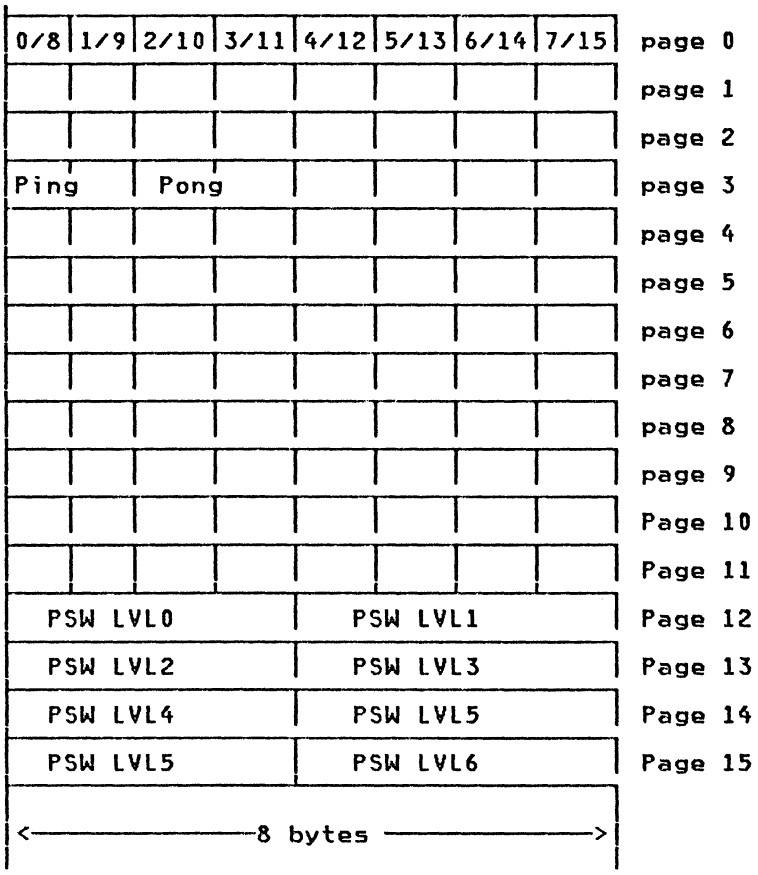
FIC is the FRU isolation code (see MIM Part 2).

CSP200 - TEST LOCAL STORE REGISTER SPACE (LSR)

The local store register space is an array of 128 entries. Each entry is an eight-bit (one byte) register which can be accessed by most of the CSP instructions. This local store is logically divided in 16 pages, each page containing 8 bytes. The last 4 pages are used by the CSP as PSWs, and the first 12 pages as register space. The test is done in two steps:

1. Test local store addressability.
2. Test pattern.

Local store Array



TEST LOCAL STORE ADDRESSABILITY

The entire local store is filled with X'FF'. The LSPR is used as follows to put its own address in each entry of the array:

X'01', X'12', X'23', X'34', X'45', X'56', X'67', X'78', X'9A', X'AB', X'BC', X'CD', X'DE', and X'EF'.

The local store addresses are X'00' through X'7F' (128 bytes).

STEP - FUNCTION:

1. Read first/next local store position, and check that the value is X'7F'. Store in it, its own address, from X'00' through X'7F'.

2. Initialize LSPR to first page and check that each local store position has its own address stored.

Error: Error found when reading local store position xx (FIC: 05).

TEST LOCAL STORE PATTERN

Two sets of pattern are used to test the capability of the local storage to retain patterns:

- First set : X'80', X'40', X'20', X'10', X'08', X'04', X'02', and X'01'.

- Second set: X'C0', X'60', X'30', X'18', X'0C', X'06', X'03', X'01', and X'00'.

FUNCTION: Initialize LSPR to first page. Store in each local store position first/next pattern.

Read back first/next pattern and check its value.

Error: Error found for pattern xx at local store position yy (FIC: 05).

ADCP0002 - TEST ADDRESS COMPARE REGISTERS

External registers XR28 and XR29 are used by the address compare function of the CSP to hold the control store address. They are tested by writing and reading back the following set of patterns:

X'FF, X'7F, X'3F, X'1F, X'0F, X'07, X'03, X'01, and X'00'.

X'AA, X'55, X'2A, X'15, X'0A, X'05, X'02, X'01, and X'00'.

FUNCTION: Write first/next pattern into XR28 and XR29.

Read back the pattern and check it.

Error: Error found for pattern xx (FIC: 05).

CSP30 - TEST CSP INTERRUPT MECHANISM

To test the interrupt mechanism of the CSP, the local store is initialized and formatted to have expected values when a change of interrupt level occurs. The table below shows how the local store is initialized, with the expected values for:

- PCI register (program controlled interrupt)
- LSPR
- CIL register (current interrupt level)

	0/8	1/9	2/10	3/11	4/12	5/13	6/14	7/15
0								
1								
2								
3	Ping		Pong					
	PCI	LSPR	CIL CIL					
4	FC	4C	18					
5	7C	5D	81					
6	3C	6D	82					
7	1C	7E	8B					
8	0C	8E	84					
9	04	9F	8D					
A	Unused		LVL6					
B	00	BC	87					
C	Level 0 PSW				Level 1 PSW			
D	Level 2 PSW				Level 3 PSW			
E	Level 4 PSW				Level 5 PSW			
F	Level 6 PSW				Level 7 PSW			

Test PCI register XR25 is an 8 bits register, one bit is dedicated to each interrupt level bit 0 if set causes an interrupt level 0 to occur, bit 1 for level 1 and so on until level 7 which is the lowest in priority. When an interrupt is requested the following registers are saved by swapping mechanism: LSPR, PSW and CCR. The routine gains control at level 0 (Level set at POR time).

FUNCTION: The interrupt request mechanism works as follows: From level 0, request a level 7 interrupt; from level 7 request a level 5 (6 is not implemented); from level 5 request a level 4 and so on through level 0. While requesting an interrupt level, the current level is not reset from the PCI register allowing a test of the priority mechanism.

The following checking is performed by each level:

- CCR value
- PCI state
- CIL and stack registers
- LSPR
- OLD PSW value

Error: Error found in testing the interrupt mechanism (FIC: 01).

TEST CSP MASKING MECHANISM

External register XR30 is the register by which interrupts can be masked in the CSP. It is an 8-bit register; bits 0 through 6 are called the common mask, and allow a selective masking of all interrupts except level 7. Bit 7 is called the master mask and can be used to mask all interrupts except level 0. Attempting to mask an interrupt at level 'n' while executing in the same level 'n' results in a NOP. The test is performed in 3 steps.

STEP - FUNCTION:

1. Test master mask for level 0 from each interrupt level:

   - set master mask.
   - Request a level 0 interrupt.
   - Check that level 0 occurred.
   - Check CIL and CIL stack.

   Error: Level 0 did not occur, or CIL and CIL stack are not equal (FIC: 05).

2. Test that the master mask bypasses all levels when the MM is set

   Check that while operating in level 0: if there is a level 'n' set in PCI and if master mask is set, when an exit is done from level 0, then level 7 gains control.

   Error: Control not given to level 7 (FIC: 01).

3. Check that no interrupt occurs when an interrupt level, including level 0, is masked via the common mask.

   Error: Masking failed for level 'n' (FIC: 01).

FIC is the FRU isolation code (see MIM Part 2).

CSP45MEM - TEST CONTROL STORE MICROINSTRUCTIONS

Two CSP microinstructions (CS0 and CS1) are used to access CSP storage. They are used with or without increment of the control store address:

- CS0: LDH, STH (Load, Store without increment), LDHI, and STHI (Load, store with increment).
- CS1: LHN, STHN (Load, Store without increment), LHNI, and STHNI (Load, store with increment).

To test these instructions, data is stored, then loaded; the control store address is checked both with and without increment.

STEP - FUNCTION:

1. **Test STH:** Local store LS0-LS1 holds data to store and LS2-LS3 holds control store address. Check data and check that the control store address did not change.

   Error: STH failed - control store address changed (FIC: 05).

2. **Test LDHI:** Displacement = X'FF' is used as LDHI operand which means: actual address is X'8FFF', and final address after LDHI execution is X'9000'.

   Error: Wrong control store address after LDHI executes (FIC: 05).

3. **Test STHI:** The displacement X'0' is used as LDHI operand, which means: Actual address is X'9000' and final address after STHI execution is X'9001'.

   Error: STHI failed to increment the control store address (FIC: 05).

4. **Test LDH:** The displacement X'0' is used as LDH operand which means:

   Actual address is X'9001';final address is X'9001'.

   Error: LDH failed - control store address is not the one expected (FIC: 05).

5. **Test STHN:** Local store LS0-LS1 is initialized to X'8FFF' (control store addr.) and LS2-LS3 to data = X'AAAA'. Check that the control store address is still X'8FFF'.

   Error: Wrong control store address after STHN executes (FIC: 05).

6. **Test LHNI:** LS0-LS1 has previous control store address. Expected address = X'9000'. Expected data = X'AAAA'.

   Error: LHNI failed to increment (FIC: 05).

7. **Test STHNI:** LS0-LS1 has previous control store address. LS2-LS3 is initialized to the pattern X'4B4B'. Expected control store address = X'9001'.

   Error: STHNI failed to increment address (FIC: 05).

8. **Test LHN:** Control store address X'9000' is used to load: Expected final address = X'9000'. Expected data = X'4B4B'.

   Error: LHN failed in data compare and in the address (FIC: 05).

FIC is the FRU isolation code (see MIM Part 2).

CSP501 - TEST CONTROL STORE

The CSP control store consists of 64K halfwords, with addresses running from X'1000' to X'FFFF'. The routine tests control store addressability (forward and backward). It also stores and reads back patterns for checking. The following patterns are used:

X'FFFF', X'AAAA', X'5555', X'1313', and X'0000'.

STEP - FUNCTION:

1. Test addressability:

   Fill up the entire control store with pattern X'FFFF' and starting from address X'1000' check that each half word location has X'FFFF'. Store control store address into actually addressed position. Do the same test backward starting from location X'FFFF' and with pattern = X'AAAA'.

   Error: Error found during addressability test (FIC: 02)

2. Test pattern:

   The following patterns are used to fill up the entire control store and are read back for checking:

   X'5555', X'1313', X'0000'.

   Error: Error found during pattern test (FIC: 02)

CSP60 - TEST ADDRESS COMPARE

External registers XR27, XR28, and XR29 are used by the CSP hardware to control the address compare mechanism. XR27 is used as a control register, and the XR28-XR29 pair is used to hold the control store address for which the compare is requested. The compare is tested for both 'data store' and 'data fetch'. When the 'compare' occurs a level 0 interrupt request is raised by the CSP.

STEP - FUNCTION:

1. Test Data Store:

   XR27 bit 5 is set on (Data Store) XR28-29 is initialized to address X'8300', and data pattern X'AA55' is stored at the above control store address.

   Check that a level 0 occurred, that XR28-29 contains X'8300', and that the data pattern is X'AA55'.

   Error: Address compare failed to occur (FIC: 05).

2. Test Data fetch:

   XR27 bit 4 is set on (Data Fetch), and XR28-29 has the previous address.

   Check that a level 0 interrupt occurred, that XR28-29 contains X'8300', and that the data pattern is X'AA56'.

   Error: Address compare failed to occur (FIC: 05).

FIC is the FRU isolation code (see MIM Part 2).

UCIF0000 - TEST CSP ERROR REGISTER XR03

The CSP error register XR03 is an external register used by the CSP to latch the origin of the error which caused a level 0 interrupt. The routine sets each bit on, from bit 0 through bit 7, and checks that each one causes a level 0 interrupt.

FUNCTION:

- Initialize the PSW for level 0.
- Set first/next bit on in XR03.
- Check that the bit has been set and that a level 0 interrupt occurred.

Error: Zero found for the test of XR03 (FIC: 03)

CSPADSP0 - TEST ADAPTER SELECTION (FES)

To test adapter selection, the routine uses the FES asynchronous path.

STEP - FUNCTION:

1. Initialize PSW level 0. Select FES (set XR04 bit 6 on)

2. Write, then read back into an FES external register (XR20), the following set of patterns: X'FF', X'7F', X'3F', X'1F', X'0F', X'07', X'03', X'01',X'00', X'AA', X'55', X'2A', X'15', X'0A', X'05', X'02', X'01', and X'00'

   Check results.

3. Address an external register (XR17) not implemented in the FES, then check that no level 0 interrupt occurs.

4. Address external register XR29 not in the FES range, then check that no level 0 interrupt occurs.

5. Suppress the adapter selection (set XR04 bit 6 off), and address an external register (XR20) in the FES range, then check that a level 0 interrupt occurred.

Error: Adapter selection test failed (FIC: 07) (FIC is the FRU isolation code (see MIM Part 2).)

UCIF9999 - TEST PARITY CHECKERS

Data transferred from the CSP1 to the CSP control store is checked for good parity on both the address (control store location) and the data. Parity checkers perform this function. The routine tests the capability of the parity checkers to detect a bad parity and to raise a level 0 interrupt request.

STEP - FUNCTION:

1. Test the parity checker for data using XR08; when this register is read, it generates bad parity.

   The following pattern set is generated with a bad parity and then written from LS to control store:

   X'FF', X'7F', X'3F', X'1F, X'0F', X'07', X'03', X'01.

   Check that level 0 occurred for each pattern, and that XR03 has bits 1 and 2 on. Restore good parity.

2. Test the parity checker for the control store address, again using XR08 to generate bad parity.

   Starting from address X'1000' to address X'FFFF' a bad parity is generated for each address; check for each address:

   - Level 0 interrupt occurs.
   - XR03 has bits 2 and 4 on.

Error: Parity checker failed to report error on bad parity (FIC: 05). (FIC is the FRU isolation code (see MIM Part 2).)

CSPNEXT - TEST ROS ADDRESS DECODE

STEP - FUNCTION:

1. Initialize the PSW for level 0.

2. Read each ROS location from address X'0000' through address X'0FFF', and checks the following:

   - At location X'0555' the data is X'5555'.
   - At location X'0AAA' the data is X'AAAA'.
   - No level 0 interrupt occurred.

Error: Error found when reading ROS location (FIC: 05). (FIC is the FRU isolation code (see MIM Part 2).)

NEXTTRN – TEST MISCELLANEOUS I/O CONTROL XR00

External register XR00 is used to control the data exchange between the CCU and the CSP.

STEP – FUNCTION:

1. Initialize PSW for level 1.

2. The routine sets each bit on, and checks that it can be both set and reset (except for bit 2 which is the 100ms timer).

   - Set first/next bit on in XR00.
   - Check that the bit is on
   - Reset the first/next bit on in XR00.
   - Check that the bit is off

3. Check that bits 0 and 1, when set on, cause a level 1 interrupt request.

Error: Error found when setting on/off condition for XR00 (FIC: 03)

CSPXR01T – TEST I/O BUS CONTROL XR01

External register XR01 is used by the CSP microcode to control the data exchange between the CCU and the CSP.

FUNCTION: The routine sets each bit on and off, and checks that it can be set and reset.

- Set XR01 first/next bit on and check that bit is on.
- Set XR01 first/next bit off and check that bit is off.

Error: Error for XR01 bit cannot be set or reset (FIC: 03)

CSPXR02 – TEST IOC BUS SERVICE REGISTER XR02

External register XR02 is mainly used by the diagnostics to control the IOC bus tags.

STEP – FUNCTION:

1. Initialize PSW for level 0.

2. The routine sets each bit on and off (except bit 4 = Halt) and checks that it can be set and reset.

   a. Set first/next bit on (bypass bit 4) and check that it is on.
   b. Set first/next bit off (bypass bit 4) and check that it is off.

3. Check that bit 1, when set on, causes a level 0 interrupt request (IOC Bus Check).

Error: Error found for XR02 bit cannot be set or reset, or level 0 did not occur for bit 1 (FIC: 03).

CSP3X000 - TEST PING AND PONG BUFFERS

The Ping and Pong Buffers are located on the CSP2 card. However, they are accessed via local store addresses 0 through 3 of local store page 2. The routine performs the test by writing and reading the following set of pattterns: X'FFFF', X'7F7F', X'3F3F', X'1F1F', X'0F0F', X'0707', X'0303', X'0101', X'AAAA', X'5555', X'2A2A', X'1515', X'0A0A', X'0505', X'0202', X'0101', and X'0000'.

The routine checks that 'Ping Busy' is set on when writing the Ping buffer, and that 'Pong Busy' is set on when writing the Pong buffer.

FUNCTION:

- Initialize PSW for level 0.
- Initialize LSPR to point to page 3.
- Write into Ping buffer first/next pattern.
- Read Ping buffer and check the data pattern.
- Write into Pong buffer first/next pattern.
- Read Pong buffer and check the data pattern.
- Check that 'Ping Busy' bit 6 of XR01 is on then reset it.
- Check that 'Pong Busy' bit 6 of XR00 is on then reset it.

Error: Error found when checking the pattern and busy condition for Ping/Pong buffers (FIC: 03)

CSPPIPO - TEST PING AND PONG BUSY

When the Ping and Pong buffers are accessed via the CS0 and CS1 microinstructions, Ping Busy (XR01 bit 6) and Pong Busy (XR00 bit 6) are set on. These bits are also set on when the Ping and Pong buffers are written using the LHR microinstruction.

The routine writes a data pattern into the Ping and Pong buffers using LHN and LHNI, and checks that 'Ping Busy' and 'Pong Busy' are set. The Ping and Pong buffers are then read back using LHN and LHNI; a check is made that 'Ping Busy' and 'Pong Busy' are set on.

The routine checks that 'Ping Busy' and 'Pong Busy' are not set when using the LR, LRI, and LHR (read) microinstructions.

STEP - FUNCTION:

1. Initialize local store LS8-9 with data pattern X'FFFF' and store it in control store hex X'8000'.

   Load Ping and Pong buffer from the above control store location using LHN and check that data pattern is X'FFFF' and that Ping and Pong Busy is set on. Do the same test with LHNI.

   Store Ping and Pong buffers into control store using STHN and STHNI and check that Ping and Pong Busy bit is set on.

   Error: LHN, LHNI, STHN, and STHNI failed to set Ping/Pong busy bit (FIC: 03)

2. Access Ping and Pong buffers (write and read) using LRI and LR, and check that Ping and Pong Busy is not set. Read Ping and Pong buffers using LHR and check that Ping and Pong Busy bit is not set on.

   Error: Ping/Pong Busy bit is set on with LR, LRI (FIC: 03)

CSPRIOTY - TEST IOC BUS TO CCU PATH (INTERNAL)

The CSP provides the capability to test the following IOC bus functions while it is disconnected from the IOC bus:

- L1 and L2 interrupts to CCU.
- Cycle steal priority.

The routine uses XR02 to test the function. L1 and L2 interrupts are latched into external register XR04 and a wrap is provided to test the logic.

STEP - FUNCTION:

1. Test L1 and L2 interrupts to the CCU.

   - Set L1 in XR05 (bit 1) and check that the bit is set on
   - Check that L1 diagnostic (XR05 bit 3) is off.
   - Set TD into XR02 bit 6.
   - Check that L1 diagnostic is on.
   - Reset TD and check that L1 diagnostic is still on.
   - Set I/O (XR02 bit 3) and check that L1 diagnostic is off.
   - Set L2 and do the same process as described above.

   Error: L1-L2 internal logic to CSP failed (FIC: 03).

2. Test cycle steal priority.

   - Set 'Channel Request' (XR01 bit 2).
   - Check that the bit is set on and that 'Channel reg' wrap (XR04 bit 7) is off.
   - Set TD and check that XR04 bit 7 is on.
   - Set 'priority high' XR05 bit 5 on and check that the bit is on.
   - Check that 'Priority Diag' XR05 bit 6 is on.
   - Reset TD.
   - Check that 'Channel Request' and 'Priority Diag' are on.
   - Reset XR01, XR02, XR05, and XR04.

   Error: Test of cycle steal priority failed (FIC: 03).

ROS IOC BUS RESPONDER

The ROS IOC bus responder communicates with the IOC bus IFT, using routines JA80 and JA81. The following functions are tested by the ROS responder when JA80 starts the communication:

- PIO (IOH) Write command.
- Level 1 interrupt request to CCU.
- Level 2 interrupt request to CCU.
- PIO (IOH) Read command.
- Get Line ID command.
- Get Error Status command.
- AIO Write, Indirect, Long command.
- AIO D/I, Long command.
- Transfer control to RAM responder.

PIO Write command:

The JA80 routine starts the communication with the ROS responder by issuing an IOH Write command with the pattern X'FFFF'. The ROS responder is dispatched at level 1 and checks the following IOC bus tags: 'I/O', 'TA', and 'Ping Busy' off. It then checks that the received data pattern is X'FFFF'. If an error is found, a level 1 interrupt request is raised and the FIC code X'03' is displayed on the hexadecimal indicators. If no error is found a level 2 interrupt request is raised. The level 1 interrupt request to the CCU is reset when JA80 issues a PIO 'Get Error Status' command. The level 2 interrupt request to the CCU is reset when JA80 issues a PIO 'Get Line ID' command.

PIO Read:

The JA80 now issues a PIO (IOH) Read command, and waits for a level 1 interrupt. The ROS responder performs Tag checking as described for the Write command, loads the data (X'0's) into the Pong buffer, and requests a level 1 interrupt to the CCU. If an error is found the responder displays FIC X'03' on the hexadecimal indicators and requests a L2 interrupt to the CCU. The level 1 and level 2 interrupts are reset as described for Write.

AIO Write, Indirect, Long:

JA80 initializes the cycle steal pointer register (local store X'3F') with the CCU address at which data is to be cycle stolen, and then issues a PIO to the responder requesting the start of the AIO operation. The responder builds a CHCW in the Ping buffer and starts the AIO, sets 'Channel Request Ready' and 'Cycle Steal Request', and exits CSP level 1. When 'Cycle Steal Grant' is sent by the CCU (IOC), a CSP level 1 is dispatched and the data read from the Pong and Ping buffers until the count is reached. Then the responder raises the EOC tag and checks the data patterns:

X'FFFF, X'0000', X'FFFF', X'B7DC', X'0269', X'B7DC', X'0269', and X'FFFF'.

If an error is found, a level 1 interrupt request is raised and FIC X'03' is displayed on the hexadecimal indicators. If no error is found, a level 2 interrupt request is raised to the CCU. The AIO described above now gets 2 more halfwords, containing the CCU address pointing to the RAM responder. This address is saved for later use.

AIO Write, D/I, Long:

The JA81 routine issues a PIO to the responder to request the AIO to be started. The responder then builds the CHCW in the Ping Buffer, and the previously saved CCU address is put in the Pong and Ping buffers respectively (2 halfwords). It then sets the CSR to the CCU and exits CSP level 1. When the CSP level 1 is dispatched due to the 'Cycle Steal Grant' from the IOC, the responder reads alternately the Ping and Pong buffers and puts the data (RAM responder) in the CSP control store starting at address X'8300'. When the count is reached, the responder raises the EOC tag and requests a level 2 interrupt to the CCU. If an error is found, the responder displays FIC X'06' on the hexadecimal indicators, and requests a level 1 interrupt to the CCU.

Note: If a RAC is displayed by JA80 or JA81, the CELIA card must be plugged to get the FIC from the hexadecimal indicators.

PA01 – FES ASYNCHRONOUS OPERATION

This routine tests the asynchronous data bus parity checker. free.

FUNCTION:

Test if the asynchronous data bus parity checker of the FES is error

If the parity is OK, the checker must raise 'adapter select acknowledgement' signal to the CSP.

If the parity is not OK, the signal is not raised. The condition must give an interrupt level 0 to the CSP with an adapter interconnection check condition.

To do this test, FES XR14 is accessed with the following values: X'00', X'FF', F'02', X'69', X'B7', and X'DC'. These values are generated with a good and with a bad parity.

Note: Bad parity generator of CSP XR08 is used to generate the bad parity.

ERC	RAC	Error Description
0611	*50	Interrupt level 0 with adapter interconnection check condition: – Occurs erroneously if an attempt is made to access the FES with good parity on the bus. – Does not occur if attempt is made to access the FES with bad parity on the bus.

PA02 – FES ASYNCHRONOUS OPERATION

This routine tests external register address selection.

STEP – FUNCTION:

1. Tests correct selection of the FES external registers.

   The following actions are done:

   - Write: XR13 with value X'0B', XR14 with value X'55', XR15 with value X'94'
   - Read and verify: XR13, XR14, XR15, and XR10 with value X'00', XR12 with value X'01', XR17 with value X'20'.
   - Write: XR15, XR14, XR13 (with same values).
   - Read and verify: XR13, XR14, XR15, XR10, XR12, XR17 (with same values).

2. Test FES type by comparing the value in CDF with the contents of XR17.

ERC	RAC	Error Description
0611	*50	One or more of the external registers verified do not contain the expected value (step 1)
0612	*FA	The value in CDF does not match the FES type read in XR17 (step 2)

PA03 - FES ASYNCHRONOUS OPERATION

This routine tests external register data validity.

FUNCTION: To test if all significant bits of writable/readable external registers can activated. For this test, only the significant bits of XR13, XR14, and XR15 are tested. XR17 and XR16 are tested at functional test time.

- XR13 patterns: X'0B'-X'34'
- XR14 patterns: X'55'-X'AA'
- XR15 patterns: X'68'-X'94'

ERC	RAC	Error Description
0611	x50	One or more of the external registers verified do not contain the expected value

PA04 - FES ASYNCHRONOUS OPERATION

This routine tests the 'odd' common bus parity checker.

FUNCTION: To test if parity checker of odd common bus is error free. This checker is activated by a read operation of an asynchronous access of FES RAM.

If parity is OK: Read operation is complete: XR16 bits 0, 1, 5 equal to 0.

If parity is not OK: Read operation is not complete: XR16 bits 0, 1, 5 equal to 1 (FES internal error).

Notes:

1. Bad parity generator of CSP XR08 is used to generate the bad parity.
2. The even common bus priority checker is tested in the synchronous mode.

ERC	RAC	Error Description
0611	x50	Read operation (complete or not) does not contain the expected value according to the test done.

PA05 - FES ASYNCHRONOUS OPERATION

This routine tests pseudo-external register area addressing.

FUNCTION: To ensure that decode of the type register (XR15) allows access to the right area of the pseudo-external register areas defined:

1. RAM A
2. RAM B
3. RAM C
4. Diagnostic pseudo-external register
5. LIC/ICC

The first register of each area is written with a specific value. The registers are read back and their values verified. The same operations are repeated in opposite order. The same operations are done again, but only on area 5 (LIC/ICC).

ERC	RAC	Error Description
0611	*50	If an error is found on the expected value for the tests exercising areas 1 through 5 and no error found in tests exercising area 5. (The other cases are detected and isolated in another routine).

PA06 - FES ASYNCHRONOUS OPERATION

This routine tests pseudo-external register addressing in RAM.

FUNCTION: To test if RAM pseudo-external register addressing is error free.

STEP:

1. Each register of RAM A is written with a specific value. The registers are then read back and the value verified. The same operations are then repeated in the opposite addressing order.

2. The same tests are repeated for RAM B and RAM C.

ERC	RAC	Error Description
0611	*50	Register does not contain expected value for RAM A or RAM B (Step 1).
0611	*50	Register does not contain expected value for RAM C (Step 2).

PA07 - FES ASYNCHRONOUS OPERATION

This routine tests pseudo-external register addressing in RAM.

FUNCTION:

To check that bits of RAM bytes can be activated. Each byte of RAM is written with the value X'00', read back, and checked. Same operations are repeated with values X'FF' and X'00'.

ERC	RAC	Error Description
0611	*50	One byte of RAM does not contain expected value for the test done

PA08 – FES ASYNCHRONOUS OPERATION

This routine tests the FES attached card address bus.

FUNCTION: To check that FES attached card address bus is error free. Some registers of some installed lines of all installed cards (LIC and ICC) are addressed (in read mode) with good and bad parity. For each parity mode, the results (acknowledge signal) are stacked and a global analysis is done by the routine.

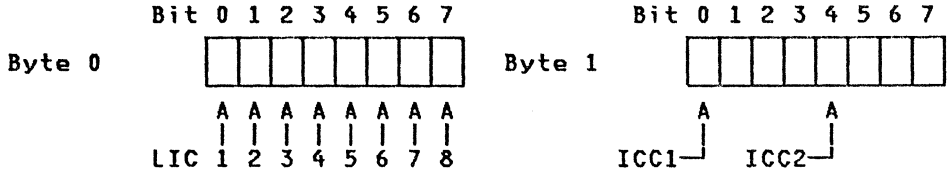
If parity OK: Read operation is complete: XR16 bits 0, 1, 4 are equal to 0.

If parity not OK: Read operation is not complete: XR16 bits 0, 1, 4, are equal to 1 (LIC/ICC error).

Note: Bad parity generator of CSP XR08 is used to generate the bad parity

ERC	RAC	Error Description
	*50	Read operation result does not contain the expected value for the test done
0611	*60	Error found in all selections (see note)
0613	*80	Error found in some selections (see note)
Note: 'ERR BIT' field gives the card (LIC/ICC) found in error		

In this routine, bits 0 through 7 of byte 0, and bits 0 and 4 of byte 1 in the ERR BIT field on the screen indicate which card is failing:



PA09 - FES ASYNCHRONOUS OPERATION

This routine tests the FES attached card data bus.

FUNCTION: To check that FES attached card data bus is error free

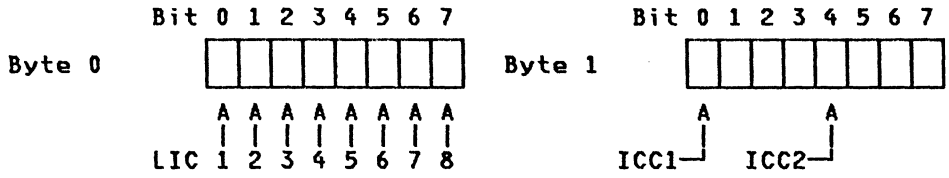
A register of the first line installed on all installed cards (LIC/ICC) are written specific patterns. Registers are read back and values verified. For ICC selection, register 10 is used. For LIC selection, register 01 is used.

Patterns used: X'0C', X'FC', and X'B4'.

Note: Bit 5 of data bus is not completely tested by that routine.

ERC	RAC	Error Description
	*50	Read data value is not the expected one for the test done
0611	*60	Error found in all selections (see note)
0613	*80	Error found in some selections (see note)
Note: 'ERR BIT' field gives the card (LIC/ICC) found in error		

In this routine, bits 0 through 7 of byte 0, and bits 0 and 4 of byte 1 in the ERR BIT field on the screen indicate which card is failing:



PA10 - FES ASYNCHRONOUS OPERATION

This routine tests the 'reset latches' command.

FUNCTION: To test if the reset latches command (FES XR17 bit 0 on) resets all latches of FES.

This command must not reset the pseudo-registers.

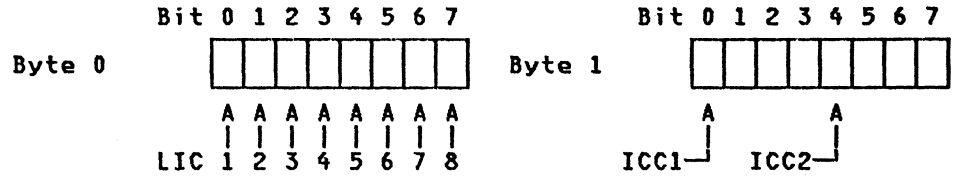
Values used: XR13 = X'3F', XR14 = X'FF', XR15 = X'BC', XR17 = X'80'.

Expected results: XR10 = X'00', XR12 = X'01', XR13 = X'00', XR14 = X'00', XR15 = X'00', XR16 = X'00', XR17 = X'20'.

Pseudo-external register must always have the value originally set.

ERC	RAC	Error Description
0611	*50	Error found on FES external value expected
0612	*50	Error found on pseudo-FES external register value

In this routine, bits 0 through 7 of byte 0, and bits 0 and 4 of byte 1 in the ERR BIT field on the screen indicate which card is failing:



PA11 - FES ASYNCHRONOUS OPERATION

This routine tests the 'reset RAM' command.

FUNCTION: To check that the RAMs are reset when the 'reset RAM' command is active (FES XR17 bit 1 on). All bytes of RAMs A, B, and C are set to X'FF'. Check that all bytes of RAMs A, B, and C are reset to X'00'.

ERC	RAC	Error Description
0611	*50	Reset not complete

PA12 - FES ASYNCHRONOUS OPERATION

This routine tests the FES attached card reset.

FUNCTION: To ensure that the 'General reset' lead, when activated from the FES, resets all the cards attached to the FES (FES XR17 bit 0 on).

One register of the first line installed of all cards (LIC and ICC) attached to the FES are initialized to a given value.

The 'General reset' lead is activated.

The register of each card is checked to be reset.

The patterns used are: LIC: X'00' in register 11, ICC: X'FF' in register 10.

ERC	RAC	Error Description
		LIC/ICC register does not contain the reset value.
0611	*60	Error found in all cards
0613	*80	Error found in some cards
Note: 'ERR BIT' field gives the card (LIC/ICC) found in error		

PA13 - FES ASYNCHRONOUS OPERATION

This routine tests the step bus parity checker.

FUNCTION: To check that the step bus parity checker is error free.

Loading the type register (XR15 bit 0-3) in the asynchronous mode allows patterns to be sent on the step bus. This operation is first performed with a good parity, then with a bad parity.

If parity is OK: Operation is complete (XR16).

If parity is not OK: Operation is not complete (XR16) (FES internal error).

The patterns used are: X'00', X'10', X'20', X'30', X'60', X'70', X'80', X'90', X'A0', X'B0', X'C0', X'D0', X'E0', and X'F0'.

Note: Bad parity generator of CSP XR08 is used to generate the bad parity.

ERC	RAC	Error Description
0611	*50	Operation result (complete or not) does not contain the expected result for the test being done.

PA14 - FES SCANNER BASE LAYER

This routine tests the scanner base layer scanning mechanism.

FUNCTION: To check that the scanning mechanism for the scanner base layer is error free.

SCF byte of all interfaces in RAM A is initialized to a given value (X'03').

Scanning is activated and a check done that interfaces to be scanned have SCF byte reset.

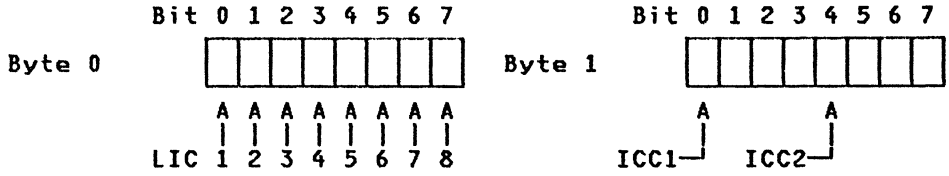
Scanning rules:

- Last LIC installed is the last LIC scanned.
- For wide band LICs and non-installed LICs (before the last LIC), only the first line is scanned.

Note: Verify also that the contents of the CDF is correct according to the machine configuartion.

ERC	RAC	Error Description
0611 0613	*60 *80	Scanning rules not met Error found in all cards (see note) Error found in some cards (see note)
Note: 'ERR BIT' field gives the card (LIC) found in error		

Note: In this routine, bits 0 through 7 of byte 0, and bits 0 and 4 of byte 1 in the ERR BIT field on the screen indicate which card is failing:



PB01 – FES SCANNER BASE LAYER

This routine tests the level 2 interrupt mechanism. It checks that a level 2 interrupt is served when the interrupt mechanism is free, and stacked in the EIRR field when the interrupt mechanism is busy.

STEP – FUNCTION:

1. Level 2 masked – stacked free. Level 2 interrupt mechanism is set up free (means no interrupt is waiting), and level 2 is masked in the CSP. An interrupt condition 1 is set on the receive interface and a condition 2 is set on the transmit interface. After scanning, check that condition 2 is stacked in the EIRR field of the transmit interface.

2. Level 2 masked – stacked busy. An interrupt condition 3 is set on transmit interface. Check after scanning that conditions 2 and 3 are cumulative in EIRR bit of the transmit interface.

3. Level 2 unmasked: Level 2 is now unmasked and a check is made at the return from the interrupt handler that the 3 conditions have raised an interrupt with the correct condition to the CSP.

ERC	RAC	Error Description
0601	*57	EIRR bit in RAM A transmit and receive do not contain expected values (Step 1).
0602	*50	EIRR bit in RAM A transmit and receive do not contain expected values (Step 2).
0603	*57	– EIRR bit in RAM A transmit and receive is not reset – Interrupt level 2 with condition 1, 2 , or 3 not raised to the CSP (Step 3).

PB02 – FES SCANNER BASE LAYER

This routine tests the Secondary Control Field (SCF) reset. It checks that all SCF fields are reset after the interface has been scanned.

FUNCTION: SCFs of transmit and receive interfaces are initialized with all significant bits set on. After scanning, check that SCF has been reset.

STEP:

1. SCF receive is initialized with X'FC'
2. SCF transmit is initialized with X'7C'

ERC	RAC	Error Description
0601	*50	SCF receive not equal to X'00' (Step 1).
0602	*50	SCF transmit not equal to X'00' (Step 2).

PB03 - FES SCANNER BASE LAYER

This routine tests the End of Burst detection for any burst size. First the transmit and then the receive interfaces are initialized, with a burst length defined by a byte count, in burst mode (with an interrupt request at the end of the burst). After the scan operation, a check is made that End of Burst was detected (interrupt raised to the CSP).

ERC	RAC	Error Description
		Interrupt (normal data transmit) not raised to CSP for:
0601	*50	End of burst with burst length = 1 byte on transmit interface
0602	*50	EOB-length 2 bytes-transmit
0603	*50	EOB-length 3 bytes-transmit
0604	*50	EOB-length 4 bytes-transmit
0605	*50	EOB-length 5 bytes-transmit
0606	*50	EOB-length 6 bytes-transmit
0607	*50	EOB-length 7 bytes-transmit
0608	*50	EOB-length 8 bytes-transmit
0609	*50	End of burst with burst length = 1 byte on receive interface
0610	*50	EOB-length 2 bytes-receive
0611	*50	EOB-length 3 bytes-receive
0612	*50	EOB-length 4 bytes-receive
0613	*50	EOB-length 5 bytes-receive
0614	*50	EOB-length 6 bytes-receive
0615	*50	EOB-length 7 bytes-receive
0616	*50	EOB-length 8 bytes-receive

PB04 - FES SCANNER BASE LAYER

This routine tests the cycle stealing of data in read mode. Check that the cycle steal data transfer (two bytes) in read mode is performed according to specific conditions.

FUNCTION: A data halfword is initialized in CSP control store. Transmit interface of RAM A is initialized with requested conditions to activate cycle steal data transfer.

After scanning, a check is made in RAM A that the cycle steal data transfer has been performed.

ERC	RAC	Error Description
0601	*55	- SPDF does not contain expected value and start interface must be reset in RAM A for for first transfer of interface (cycle steal activated).
0602	*50	- SPDF does not contain expected value for normal transfer (cycle steal activated).
0603	*50	- SPDF does not contain expected value for end of underrun (cycle steal activated).
0604	*50	- SPDF does not contain expected value for underrun detection (cycle steal is not activated).

PB05 - FES SCANNER BASE LAYER

This routine tests cycle steal of data in write mode (one-byte transfer). It checks that the cycle steal data transfer (one byte) in write mode is performed according to specific conditions.

FUNCTION: Receive interface of RAM is initialized with conditions requested to activate cycle steal data transfer (for one byte: PDF pointer off).

After scanning check that in the CSP control store the cycle steal data transfer (one byte) has been performed.

ERC	RAC	Error Description
		CSP control store does not contain the expected value for:
0601	*55	Transfer for modem change
0602	*50	Transfer for end of burst
0603	*50	Transfer for end 1 condition
0604	*50	Transfer for end 3 condition
0605	*50	Transfer for end of overrun
0606	*50	No transfer at overrun detection

PB06 - FES SCANNER BASE LAYER

This routine tests cycle steal of data in write mode (two-byte transfer). It checks that the cycle steal data transfer (two bytes) in write mode is performed according to specific conditions.

FUNCTION: Receive interface of RAM is initialized with the conditions requested to activate cycle steal data transfer (for two byte transfer: PDF pointer on).

After scanning, a check is made in the CSP control store that the cycle steal data transfer (two bytes) has been performed and that the PDF pointer in FES is reset.

ERC	RAC	Error Description
0601	*50	- CSP control store does not contain expected value. - PDF pointer in FES must be off

PB07 - FES SCANNER BASE LAYER

This routine tests the cycle steal of the status and parameter zone (transmit interface). It checks that the cycle steal of the status and parameter zone on the transmit interface are performed according to specific conditions. It also checks status and parameter validity.

FUNCTION: Transmit interface of RAM A is initialized with the conditions to activate cycle steal status transfer of one burst and/or cycle steal parameter transfer of same/next burst. The following are checked:

- Status and parameter transfer,
- Level 2 interrupt occurs
- Burst address update in address field of RAM A.

ERC	RAC	Error Description
	*50	- CSP control store param/status does not contain expected value. - FES parameter area does not contain expected value. - FES cycle steal address area does not contain expected value. - Interrupt level 2 event does not contain the expected result for the test done. For the following:
0601		Normal end of burst.
0602		End of burst with MCC remembrance.
0603		End of burst with end of message (EOM).
0604		End of burst with transmit continuous.
0605		End of burst with SYN insert.
0606		End of Transmission.
0607		Normal modem change.
0608		Modem change with start.
0609		Modem change with SYN insert.
0610		Modem change with burst not valid.
0611		Modem change direct.
0612		Underrun detection (without TE).
0613		Underrun permanent status (without TE).
0614		End of underrun (without TE).
0615		Underrun Detection (with TE).
0616		Underrun permanent status (with TE).

PB08 - FES SCANNER BASE LAYER

This routine tests the cycle steal of the status and parameter zone (receive interface). It checks that the cycle steal of the status and parameter zone on the receive interface are performed according to specific conditions. It also checks status and parameter validity.

ERC	RAC	Error Description
0601	*50	Ending condition 1+2 without EP.
0602	*50	Ending condition 2+3 without EP.
0603	*50	Ending condition 1+3 without EP.
0604	*50	Ending condition 1+2+3 (force 0).
0605	*50	Ending condition 1+2 with EP.
0606	*50	Ending condition 2+3 with EP.
0607	*50	Ending condition 1+3 with EP.
0608	*50	End of burst.
0609	*50	Modem change detection.
0610	*50	Modem change with burst not valid.
0611	*50	Modem change direct.
0612	*50	Overrun detection.
0613	*50	End of overrun.
0614	*50	Overrun with ending condition (e1).
0615	*50	Overrun without ending condition.

PB09 - FES SCANNER BASE LAYER

This routine tests the next halfword address update (bits 14 and 15 in the RAM A address field). It checks that the cycle steal address is incremented by one when the correct conditions are met.

FUNCTION: Conditions are met to reach halfword boundary in burst processing on both transmit and receive interface.

Checked that the address field in RAM A is incremented by one.

The part of the address tested by this routine is only bits 14 and 15 (halfword count).

ERC	RAC	Error Description
	*50	- FES RAM A address field bits 14 and 15 do not contain expected value. - FES RAM A PDF pointer is not reset. For:
0601		Address update from 00 to 01 on transmit.
0602		Address update from 01 to 10 on transmit.
0603		Address update from 10 to 11 on transmit.
0604		Address update from 11 to 00 on transmit.
0605		No end of burst with PDF pointer off on transmit.
0606		No end of burst with PDF pointer off and 'Start on Odd' on transmit interface.
0607		Address update from 00 to 01 on receive.
0608		Address update from 01 to 10 on receive.
0609		Address update from 10 to 11 on receive.
0610		Address update from 11 to 00 on receive.
0611		No end of burst with PDF pointer off on receive.

PB10 - FES SCANNER BASE LAYER

This routine tests the next burst address update. It checks that the burst address in FES RAM A address field bits 9 to 13 is updated to the next burst address when a burst change occurs.

FUNCTION: Both transmit and receive interfaces are initialized with a burst change condition.

After scanning, check that the address field in RAM A has been updated to the next burst address. This mechanism uses 'buffer length parameter' set in RAM A.

The address bits tested by this routine are 9 to 13 of the address field depending on bits 0 and 1 value.

ERC	RAC	Error Description
0601	*50	FES RAM A - Buffer length field - Address field does not contain the expected value for:
0601		Address field update on transmit.
0602		Address field update on receive.

PB11 - FES SCANNER BASE LAYER

This routine tests data byte transfer to the front end (transmit). It checks that the data byte is sent to the front end layer from the SPDF of RAM A or directly from the CSP control store.

FUNCTION: Two halfword patterns are initialized in the CSP control store at beginning of burst. Transmit interface of RAM A is initialized in order to perform write front end phase following cycle steal data phase. Checked that in RAM B the even byte or odd byte, depending on start on odd condition and position of halfword in burst, has been sent directly to front end. The write front end is performed without a cycle steal data. A check is made that SPDF in RAM A has been moved to PDF of RAM B.

ERC	RAC	Error Description
0601	*50	Direct transfer with start on odd at beginning of the burst. - PDF in RAM B does not contain expected value.
0602	*50	Direct transfer without start on odd at beginning of the burst. - PDF in RAM B does not contain expected value. - SPDF in RAM A does not contain expected value. - PDF pointer in RAM A does not contain the expected value.
0603	*50	Normal direct transfer - PDF in RAM B does not contain expected value. - SPDF in RAM A does not contain expected value.
0604	*50	Data transfer from SPDF - PDF in RAM B does not contain expected value.

PB12 - FES SCANNER BASE LAYER

This routine tests control byte transfer to the front end. It checks the transfer and the validity of the control byte field sent from the scanner base to the front end.

FUNCTION: Transmit interface on RAM A is initialized in order to perform data byte transfer from scanner base to front end. A check is made in RAM B that the control byte field associated with the data byte has been transferred to the front end with valid data. The receive interface is initialized in order to perform a control byte transfer from scanner base to the front end. Same checks are performed as for transmit.

ERC	RAC	Error Description
		RAM B control field does not contain the expected value and RAM A parameter field do not contain the expected value for:
0601	*50	- Direct transfer at beginning of burst on transmit.
		RAM B control field does not contain the expected value for:
0602	*50	- Transfer from SPDF during normal burst processing on transmit (signal NOZI on).
0603	*50	- Transfer from SPDF during normal burst processing on transmit (signal NOZI off).
0604	*50	- Transfer from SPDF at end of burst on transmit (signals on).
0605	*50	- Transfer from SPDF at end of burst on transmit (signals off).
0606	*50	- No transfer for underrun detection on transmit.
0607	*50	- No transfer for force 10 timer full on transmit.
		RAM B control field does not contain the expected value and RAM A parameter field do not contain the expected value for:
0608	*50	- Direct transfer at beginning of burst on receive.
		RAM B control field does not contain the expected value for:
0609	*50	- Normal transfer on receive
0610	*50	- No transfer for overrun detection on receive.

PB13 - FES SCANNER BASE LAYER

This routine tests byte stacking on the receive interface.

FUNCTION: Check that NPDF byte is stacked in SPDF byte when conditions are met.

PDF pointer is set on after stacking.

ERC	RAC	Error Description
0601	*50	- SPDF field in RAM A does not contain the expected value. - PDF pointer in RAM A does not contain the expected value.

PB14 - FES SCANNER BASE LAYER

This routine tests the asynchronous timer. It checks that the timer works correctly in asynchronous mode.

FUNCTION: Transmit and receive interfaces are initialized in order to start the timer.

Scanning is activated and a check is made after a clock time that the timer has been incremented. It is also checked that timer full is detected when the condition is met.

ERC	RAC	Error Description
0601	*68	Short timer with direct starting (start testing). - Timer value in RAM A receive does not contain expected value. - Interrupt level 2 with timeout must not be done to the CSP. Note: First detection of 480 hz grounded in TSS.
0603	*50	Short timer with direct starting (increment testing). - Interrupt level 2 with timeout must not be done to the CSP.
0604	*50	Short timer with direct starting (timer full). - Interrupt level 2 with timeout must not be done to the CSP. - Timer value in RAM A must be reset.
0605	*50	Short timer with indirect starting at burst boundary (start timer function). - Timer work bit must be on in RAM A receive - Timer value in RAM A receive does not contain expected value
0606	*50	Short timer with indirect starting at end of transmission (start testing). - Timer work bit must be on in RAM A receive
0607	*50	Short timer with indirect starting at end of transmission with turn around. - Timer work bit must be on in RAM A receive
0608	*69	Long timer with direct starting (start testing) - Interrupt level 2 with timeout must not be done to the CSP. - Timer value in RAM A receive does not contain expected value Note: First detection of 100ms grounded in TSS.
0609	*50	Long timer with direct starting (timer full testing) - Timer value in RAM A must be reset.

PB15 - FES SCANNER BASE LAYER

This routine tests the synchronous timer (force 10). It checks that timer force 10 is correctly handled depending on specific conditions.

FUNCTION: Transmit and receive interfaces of RAM A are initialized in order to initialize, activate, and stop the timer force 10 respectively.

A check is made that timer works correctly in each case.

ERC	RAC	Error Description
0601	*50	Force 10 initialization - Timer value in RAM A receive does not contain expected value. - Timer working associated bits does not contain the expected state.
602	*50	Force 10 timer full (without TE) - Timer working associated bits does not contain the expected state. (but timer continues to work).
603	*50	Force 10 timer full (with TE) - Timer working associated bits does not contain the expected state. (but timer full always stacked).
604	*50	Force 10 reset. EOT must stop and reset the synchronous timer used for BSC transmission. - Timer value and working associated bits expected to be 0

PB16 - FES SCANNER BASE LAYER

This routine tests the synchronous timer (force 30 - force 0). It checks that timer force 30 is correctly handled depending on specific conditions. It also checks that the timer is reset when the force 0 conditions are met.

FUNCTION: Receive interface of RAM A is initialized in order to initialize, activate and stop the timer force 30.

- A check is made that timer works correctly in each case.
- A check is also made that timer is reset when force 0.

ERC	RAC	Error Description
0601	*50	Force 30 initialization - Timer work bit must be on in RAM A receive. - Timer value in RAM A does not contain the expected value.
0602	*50	Force 30 timer full - Interrupt level 2 with timeout condition must be done to the CSP.
0603	*50	Force 0 - Timer must be stopped and reset.

PB17 - FES SCANNER BASE LAYER

This routine tests three address control in SDLC mode. It checks that the three address control condition is detected on the receive interface at parameter transfer.

FUNCTION: On receive interface, cycle steal status is performed because of 'flag OK' (end 3) condition on burst 'n'. Parameter of burst 'n+1' is transferred with the 'three address control condition'.

Check that the burst length in RAM A is forced to the 3 bytes value in order to isolate the 'one address and two controls' or 'two addresses and one control' in one burst.

ERC	RAC	Error Description
0601	*50	Three addresses control after end 3 flag. - RAM A burst length must be equal to 3 bytes.
0602	*50	Three addresses control without a previous end 3 flag (it is not a three addresses control). - RAM A burst length must not be changed.

PB18 - FES SCANNER BASE LAYER

This routine tests the even common bus parity checker. It checks that the parity checker of the even common bus is error free.

FUNCTION:

- Transmit interface is initialized in order to perform a write front end phase. SPDF is loaded with good and bad parity. A check is made that FES internal error level 2 is raised when parity is bad.
- Receive interface is initialized in order to stack the NPDF in the SPDF. NPDF is loaded with good and bad parity. A check is made that FES internal error level 2 is raised when parity is bad.

Note: Bad parity generator of CSP XR08 is used to generate the bad parity.

ERC	RAC	Error Description
0601	*50	Good parity on transmit interface. Interrupt level 2 FES internal error must not be done to the CSP
0602	*50	Bad parity on transmit interface. Interrupt level 2 FES internal error must be done to the CSP
0603	*50	Good parity on receive interface. Interrupt level 2 FES internal error must not be done to the CSP
0604	*50	Bad parity on receive interface. Interrupt level 2 FES internal error must be done to the CSP

PB19 – FES SCANNER BASE LAYER

This routine tests error reporting in synchronous mode. It checks that parity errors are detected and reported in the EIRR field of the interface under test when the FES is running in synchronous mode.

FUNCTION: Both receive and receive interfaces are initialized in order to perform cycle steal status and cycle steal data (for receive only) with bad parity.

A check is made that errors are correctly reported in the EIRR field.

Note: Bad parity generator of CSP XR08 is used to generate the bad parity.

ERC	RAC	Error Description
0601	*50	Bad status transfer on transmit Level 2 – CSP/FES error must be raised to the CSP.
0602	*50	Bad status transfer receive Level 2 – CSP/FES error must be raised to the CSP.
0603	*50	Bad status transfer receive Level 2 – CSP/FES error must be raised to the CSP. Level 2 – FES internal error must not be raised to the CSP.

PC01 - FES FRONT END LAYER

This routine tests the Modem-Out driver check function.

ERC	Function	Error Description	RAC
0012	Driver check detection and reporting on Modem-Out signals. 1- Test if no driver check reporting when it is not necessary. 2- Test if driver check is reported when error on Modem-Out bit 4. Repeat the test on Modem-Out bits 3, 2, 1, and 0.	 1- Interrupt level 2 with driver condition occurs erroneously. 2- No interrupt level 2 with driver check condition.	*58 (Note)
0013	Transmit bit (Data bit) driver check reporting and masking.	No interrupt level 2 with driver check condition.	*58
0014	Driver check masking on Modem-Out signals. Test if no driver check is reported when error occurs on Modem-Out bit 4 then repeat for bits 3, 2, 1, and 0, but with corresponding bit in error masked.	Interrupt level 2 with driver check condition occurs erroneously.	*58

Note: FES in diagnostic mode using modem driver check facility

PC02 - FES FRONT END LAYER

This routine tests Modem Change Detection and Modem-Out Sending.

ERC	Function	Error Description	RAC
0016	FES Modem Change Detection and FES Modem-Out Sending functions. 1) Test if modem change is not detected when there is no change on Modem-In set to all 1s. Repeat with bits 0, 1, 2, 3, and 4 of Modem-In set to 0. 2) Test if modem change is detected when a change (from 1 to 0 or 0 to 1) occurs on bits 5, 4, 3, 2, 1, and 0 of Modem-In. 3) Test if modem change function is stopped in FES when a change is already detected. 4) Test if Send Modem-Out Stacked work correctly. Tested value = X'A8' and X '50'.	3 conditions signal a modem change detection: - Modem change stopped and new modem value is saved in Modem-In. - Modem change condition is signaled on receive interface. - Modem change condition is signaled on the transmit interface. 1) One (or more) conditions not found. 2) One (or more) conditions not found. 3) One (or more) conditions not found. 4)- Modem-Out Send ≠ Modem-Out Stacked - Modem-Out Immediate ≠ Modem-Out stacked.	*53 (Note)

Note: FES in diagnostic mode using Modem-Out Modem-in Wrap facility

PC03 - FES FRONT END LAYER

This routine tests Modem Change Masking and No Modem Change reporting on the receive interface.

ERC	Function	Error Description	RAC
0017	Modem Change Masking: Test if modem change is not detected when a modem change occurs but with the corresponding signals masked.	One (or more) of the 3 conditions defining a modem change detected are on (see ERC PC020016).	*53 (Note)
	The test is done with: - all bits changed (from 1 to 0 and 0 to 1) and all bits masked. - bit 4 then 3, 2, 1, 0, is changed (from 1 to 0 and 0 to 1) and the corresponding bit masked (Note: FES in diagnostic mode using Modem-out Modem-in Wrap facility)		
0018	No modem change reporting on receive interface. Test that when a modem change is detected, it is not reported on the receive interface.	- The modem change is not detected. - The modem change is reported on receive interface.	*53

PC04 - FES FRONT END LAYER

This routine tests start data management on the transmit interface.

ERC	Function	Error Description	RAC
0001	Start data management on transmit interface. Start bit set on in scanner base layer parameter area of a Transmit interface must start the data management on the corresponding front end layer transmit interface if a modem change is not locked on the interface (start delayed by front end layer). 1) The test is done without Modem Change locked. 2) The test is done with a Modem Change locked.	3 criteria for start to be effective: - GAD bit in RAM C = 1 - Start bit in RAM B = 1 - Start bit in RAM A = 0 3 criteria for start delayed: - GAD bit in RAM C = 0 - Start bit in RAM B = 1 - Start bit in RAM A = 0 1) One (or more) of 3 criteria for start effective not found 2) One (or more) of 3 criteria for start delayed not found	*53 (Note)
0003	Start data management on receive interface. On the receive interface the modem change condition is ignored. The test is done with and without modem change.	One (or more) of 3 criteria for start effective not found	*53 (Note)
Note: FES in diagnostic mode			

PC05 - FES FRONT END LAYER

This routine tests synchronous error reporting.

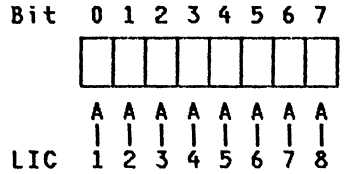
ERC	Function	Error Description	RAC
0601	Synchronous error reporting LIC/ICC check. Test of FES mechanism reporting LIC/ICC check.	The following conditions must occur: - Level 2 LIC Check on Transmit interface - Data management stopped on Transmit interface. - Level 2 LIC and ICC check on Receive interface. - Data management stopped on Receive interface	*58
	Note: FES in diagnostic mode using error detection checking mechanism		
0602	Synchronous error reporting. FES internal error.	Following conditions must occur: Level 2 front-end internal error - Data management stopped on corresponding interface.	*58 (Note)
	FES internal error must be reported when a bad parity is found on the even byte by front end layer.		
0603	Same as 602 but on odd byte.	As above. (Note)	*58
Note: FES in diagnostic mode using CSPs bad parity facility			

PC06 - FES FRONT END LAYER

This routine tests front end scanning: it checks the 'LIC present' and 'LIC wide band' lines between the FES and the LICs.

ERC	Function	Error Description	RAC
0501	Front end layer scanning is tested by a comparison between the scanner configuration (given by CDF) and scanning result (using propagation of front end start data).	One (or more) interfaces attached to the same LIC are not correctly scanned.	*86 (Note)
		Several interfaces attached to several LICs are not correctly scanned (the LIC positions found in error are flagged by the ERRBIT field.	*65 (Note)
	The 64 interfaces are tested, and, according to the result, an error message is issued: - If an interface is not scanned when the CDF and scanning rules show the interface to be present. - If an interface is scanned when the CDF and scanning rules show the interface not present.		
Note: Verify also if the contents of the CDF is correct			

In this routine, the first 8 bits of the ERR BIT field on the screen indicate which card is failing:



A LIC card presence will only be tested if the CDF shows at least one line installed on the LIC card.

PC07 - FES FRONT END LAYER

This routine tests data management in Start/Stop mode.

ERC	Function	Error Description	RAC
0100	S/S transmit - normal data management. Transmission test done in: S/S 8 bits, 1 stop bit. S/S 8 bits, 2 stop bits. S/S 7 bits, 2 stop bits. S/S 6 bits, 1 stop bit. S/S 5 bits, 2 stop bits.	One (or more) of following condition is not correct: -Start bit generation (= 0). -Serialization according to character length and value. -Stop (1 or 2) bit generation (= 1) -Status correctly set by FES in CSP control store.	*5D (Note)
0101	S/S transmit break function	Data bits sent or status not correct	*5D
	(stop bit(s) are generated at 0 (instead of 1). Transmission test done in: S/S 8 bits, 1 stop bit - S/S 7 bits, 2 stop bits		
0102	S/S Xmit Underrun function.	Data bits sent not correct or level 2 with underrun condition not sent to CSP.	*58 (Note)
	When underrun is detected by front end in S/S, only bits at 1 are transmitted. Transmission test done in: S/S 8 bits, 1 stop bit - S/S 6 bits, 2 stop bits		
0103	S/S transmit - EOM request. When EOM is requested, at the end of the corresponding burst, the FE generates 3 bits at 1, sends interrupt level 2 with EOM condition, and stops data management on the corresponding interface.	One (or more) of following conditions is not correct: - Data bits sent on line. - Burst status in CSP. - Data management not stopped. - Level 2 interrupt (with EOT condition) not done to CSP.	*5D (Note)
0104	S/S Receive normal data management. Reception test done in: S/S 8 bits, 1 stop bit S/S 8 bits, 2 stop bits S/S 7 bits, 2 stop bits S/S 6 bits, 1 stop bit S/S 5 bits, 1 stop bit	One (or more) of following conditions is not correct: - All bits before start bit (= 0) are deleted - Start bit deletion. - Character assembly according to character length and value - Stop bit deletion according to stop length. Receive status correctly set by FES In CSP control store.	*5D (Note)
0105	S/S Receive - Stop Check function. Stop bit(s) of a Start/Stop character must always be at 1.	- Result of data character receive not correct, or - Status set by FES in CSP control store not correct according to stop bits received.	*5D (Note)
	The test is done in: S/S 8 bits, 1 stop bit (stop bit not OK and OK). S/S 7 bits, 2 stop bits (first stop bit not OK and OK). S/S 5 bits, 2 stop bits (second stop bit not OK and OK). Note: FES in diagnostic mode using bit injection facility		
0107	Start/stop transmit Start bit = mark	Data bits transmitted or status not correct.	*5D (Note)
	When start bit at mark is requested, the start bit generated for the last character of the burst is equal to 1 instead of 0. Test is done according to character length.		
Note: FES in diagnostic mode using bit sample facility			

PC08 - FES FRONT END LAYER

This routine tests the overrun function in Start/Stop receive mode; it also tests Modem Change reporting.

ERC	Function	Error Description	RAC
0106	Overrun function in S/S Receive.	One overrun condition not found	*5F
	When the FES detects a burst not valid on a receive interface: - An interrupt level 2 (with overrun condition) is sent to the CSP. - Write of data burst is not done. - When the burst becomes valid the writing of data burst restarts and the overrun condition is set in the next status. Note: FES in diagnostic mode using bit injection facility		
0020	Modem Change reporting on receive interface.	One condition not found on the receive interface.	*5F
	When a Modem Change is detected by the FES, the reporting of the modem change is done by: - An interrupt level 2 with modem change condition. - The modem change signal is set on in the status. This reporting is however delayed until the first character boundary (test is done at character boundary or not). Note: FES in diagnostic mode using modem-out modem-in wrap facility and bit injection facility (test done using S/S protocol)		
0022	Modem Change reporting on transmit interface.	One condition not found on transmit interface.	*5F
	Same test as for ERC 020 but the conditions (interrupt and status) are on the transmit interface. Note: FES in diagnostic mode using modem-out modem-in wrap facility and bit sample facility		

PC09 - FES FRONT END LAYER

This routine tests data management in SDLC transmit mode.

ERC	Function	Error Description	RAC
0201	SDLC transmission (test of the No Zero Insert (NOZI) function. 1- With NOZI function off, test if a 0 is inserted after 5 consecutive 1s.	One (or more) of following conditions is not correct: 1- Data bits sent on line. - Burst Status in CSP.	*5D (Note)
	This test is done for a character, for first bit of a character, at a character boundary, and between two characters.		
	2- With NOZI function on, test if no 0 is inserted after 5 consecutive ones, and if BCCs are preset to all 1s.	2- Data bits sent on line. - Burst status in CSP. - BCCs not at 1s.	
0202	SDLC transmission. Test of Send CRC function.	- Data bits sent on line. - Burst status in CSP.	*5D (Note)
	Test if BCCs are correctly transmitted when requested, and verify that the zero insert function works correctly on BCC transmission.		
0203	SDLC transmission. Test of Non-Return to Zero Inverted (NRZI) function.	- Data bits sent on line. - Burst Status in CSP.	*5D (Note)
	When ON, the NRZI function modifies the output data bit according to the algorithm: Output data bit = XOR inverted between last line state (LLS) and data bit to be sent.		
0204	SDLC transmission. Test of CRC accumulation.	- Data bits sent on line. - Burst Status in CSP.	*5D (Note)
	BCCs are calculated using the following algorithm: $X^{16} + X^{12} + X + 1$ Transmission of a special pattern (in SDLC) is used to completely test this algorithm.		
0205	SDLC transmission. Test of OEM processing.	- Data bits sent on line. - Burst Status in CSP. - Level 2 interrupt with EOT condition EOT condition. - Data management stopped.	*5D (Note)
	When EOM is requested: - at the end of corresponding burst, the front end generates 3 bits at 1, ignoring NRZI function. - Do an interrupt level 2 with EOM condition. - Stop data management on corresponding interface.		
Note: FES in diagnostic mode using bit sample facility			

PC10 - FES FRONT END LAYER

This routine tests data management in SDLC receive mode.

ERC	Function	Error Description	RAC
0207	SDLC Receive. Test of No Zero Delete function. When option is off, after 5 consecutive 1s, if there is a 0 it is deleted. When option is on, the 0 is not deleted.	- Data burst receive. - Burst status in CSP.	*5D (Note)
0208	SDLC Receive. Test of NRZI function. When on, the NRZI function modifies the input bit according to the algorithm: Input Data Bit = XOR inverted between last line state (LLS) and data bit received.	- Data burst receive. - Burst status in CSP.	*5D (Note)
0209	SDLC Receive Flag Processing Following cases of flag are tested: - Synchronization flag without interrupt on first flag option. - Synchronization flag with interrupt on first flag option. - Flag to test BCCs with BCCs OK. - Flag to test BCCs with BCCs not OK. - Flag not at character boundary.	- Data Burst Receive. - Burst Status receive in CSP. - BCC value in front end not correct - Interrupt level 2 (with ending flag condition) not sent to CSP.	*58 (Note)
0210	DLC Receive Abort Processing. The following cases of abort are tested: - Abort detect at a character boundary. - Abort detect not at a character boundary.	- Data burst receive. - Burst Status Receive in CSP. - BCC value in front end not correct - Interrupt level 2 (with ending flag condition) not done to CSP.	*58 (Note)
Note: FES in diagnostic mode using bit injection facility			

PC11 - FES FRONT END LAYER

This routine tests data management in SDLC receive mode, and the underrun process in SDLC transmission.

ERC	Function	Error Description	RAC
0211	SDLC Receive. Idle Processing. Following cases of idle are tested: - Idle detected at a character boundary. - Idle detected out of a character boundary.	- Data burst receive. - Burst Status Receive in CSP. - BCC value in front end not correct. - Synchronization not stopped. stopped. - Interrupt level 2 (with ending flag flag condition) not done to CSP.	*5A (Note)
0212	SDLC Receive = CRC function.	- Data burst receive. - Burst status Receive in CSP. - BCCs value in FE not correct.	*5F (Note)
	BCCs are calculated with the following algorithm: $X^{16} + X^{12} + X + 1$ The final BCC value must be equal to X 'F0B8'. Test is done with a receive bit pattern allowing to test the algorithm and with a final BCC OK, then with a final BCC wrong in bit 0, then with a final BCC wrong in bit 1, and so on up to bit 15.		
0206	SDLC transmission: underrun Processing.	- Data bits sent on line. - Interrupt level 2 (with underrun condition) not sent to CSP. - Only the restart of the interface is effective to exit underrun condition.	*5F (Note)
	When underrun is detected in SDLC, the front end layer generates an abort character and sends FLAG, FLAG, ... continuously until the restart of the corresponding interface. An interrupt is raised to the CSP. Note: FES in diagnostic mode using bit sample facility		
Note: FES in diagnostic mode using bit injection facility			

PD12 - FES FRONT END LAYER

This routine tests data management in BSC transmission:

- Normal transmission in BSC Coding.
- Underrun process in BSC Coding.

The function starts in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.

ERC	Function	Error Description	RAC
0301	BSC Coding Transmit. Normal data management. Transmission test done in: BSC coding 8 bits BSC coding 7 bits BSC coding 6 bits	- Data bits sent on line. - Burst Status Transmit in CSP.	*5D (Note)
0302	BSC Coding Transmit Underrun process. When underrun is detected by the FE in BSC coding, the FE layer generates DLE-SYN (defined in FES RAM) continuously until the underrun exit. The test is done in: BSC coding 8 bits, BSC coding 7 bits, BSC coding 6 bits.	- Data bits sent on line.	*5D (Note)
0304	Start function in BSC transmission. Must start the corresponding interface in BSC control mode. Test done in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.	- Control mode defined in front end working bits. - Burst Status Transmit in CSP.	*53 (Note)
Note: FES in diagnostic mode using bit sample facility			

PD13 - FES FRONT END LAYER

This routine tests data management using BSC transmission in the control mode. All tests are done in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.

ERC	Function	Error Description	RAC
0305	BSC Control mode transmission BSC control character processing in control mode	- Data bits sent on line. - Burst Status Transmit in CSP. - Control mode defined in front end working bits.	*5D (Note)
	All control characters (except STX and SOH) are normally transmitted to the line. Test done for: DLE-SYN-ITB- ETB-ETX-ENQ-EOT-NAK-ACK0- ACK1-WACK-RVI.		
0306	BSC control mode transmission. No BCC is generated after ITB or ETB or ETX in control mode, nor does CRC or LRC accumulation take place in control mode.	- Data bits sent on line. - Burst status Transmit in CSP. - BCCs or LRC in FES must be equal to initial value according to BSC and CRC type.	*5D (Note)
0307	BSC control mode transmission. Test of VRC generation.	- Data bits (with VRC bits) sent on line - Burst Status Transmit in CSP.	*5D (Note)
	VRC generation works normally in control mode. Test done in BSC ASCII 7 Bits.		
0308	BSC control mode transmission. Test of STX and SOH processing.	- Data bits sent on line. - Normal mode defined in front end working bits. - BCCs or LRC in FES must equal to initial value according to CRC type	*5D (Note)
	When STX or SOH is detected, the front end layer enters the BSC normal In the CRC B-LRC-CRC S, the STX character is not accumulated in the BCCs.		ode.
0309	BSC control mode transmission. Test of DLE STX sequence processing.	- Data bits sent on line. - Transparent mode defined in front working bits - BCCs or LRC in FES must be equal to initial value according to CRC type	*5D (Note)
	When DLE STX is detected the front end layer enters the BSC transparent mode. In the CRC B-LRC-CRC S, the STX character is not accumulated in the BCCs.		
0310	BSC control mode transmission. Test of STX accumulation in CRC.	- Data bits sent on line. - Normal mode or transparent mode is defined in front end working bits. - BCCs value = value of STX character accumulated according to BSC/CRC type defined.	*5D (Note)
	The STX character allowing entry to the normal mode or the transparent mode (with DLE-STX sequence) is accumulated in the CRC, according to BSC type, when the CRC type = 00 (STX included).		
Note: FES in diagnostic mode using bit sample facility			

PD14 - FES FRONT END LAYER

This routine tests data management using BSC transmission in normal mode, and also tests SYN-SYN generation.

ERC	Function	Error Description	RAC
0311	BSC normal mode transmission Test of SYN-SYN generation	- Data bits sent on line. - Burst Status Transmit in CSP. - level 2 interrupt with underrun condition. - Level 2 interrupt with timeout condition.	*5D (Note)
	- When underrun is detected by FE in BSC normal mode, it generates a SYN-SYN sequence continuously (doing an interrupt level 2 with underrun) until underrun exit. - Every second (timer full) if the front end is in the normal mode and the option SYN Insert is on, the FE generates a SYN-SYN sequence and continues normally. If the option SYN Insert is off, the FES does an interrupt level 2 with a timeout condition. The test is done in: BSC EBCDIC, Test done in: BSC ASCII 8 bits.		
0312	BSC normal mode transmission	- Data bits sent on line. - Burst status Transmit in CSP.	*5D (Note)
	Test if SYN-SYN generation is delayed by the front end when working bit TE is on (used to send a blocked sequence). The test is done in: BSC EBCDIC.		
Note: FES in diagnostic mode using bit sample facility			

PD15 - FES FRONT END LAYER

This routine tests data management using BSC transmission in normal mode. All tests are done in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.

ERC	Function	Error Description	RAC
0314	BSC normal mode transmission DLE decoded When DLE is decoded in the normal mode, the TE bit is set on to indicate a locked sequence of characters (if necessary, SYN insert is delayed).	- Burst status Transmit in CSP (with the TE bit on).	*5D (Note)
0315	BSC normal mode transmission ITB decoded When ITB is decoded in the normal mode, the TE bit is set on to indicate a locked sequence of characters (if necessary, SYN insert is delayed).	- Burst status Transmit in CSP (with the TE bit on).	*5D (Note)
0316	BSC normal mode transmission ENQ decoded When ENQ is decoded in the normal mode, the TE bit is set on to indicate a locked sequence of characters (if necessary, SYN insert is delayed).	- Burst status Transmit in CSP (with the TE bit on).	*5D (Note)
0317	BSC normal mode transmission SYN-SYN character generation SYN-SYN characters generated by the front end are not accumulated in CRC in normal mode. The test done is CRC B-LRC-CRC S.	- BCCs or LRC in FES RAM must be equal to initial value according to CRC type and BSC type	*5D (Note)
0318	BSC normal mode transmission SYN-SYN characters sent in a message. SYN-SYN characters sent in a message (as data) are not accumulated in CRC in normal mode. The test done is CRC B-LRC-CRC S.	- BCCs or LRC in FES RAM must be equalto initial value according to CRC type and BSC type.	*5D (Note)
0319	BSC normal mode transmission An ENQ character decoded in normal mode allows a return to the control mode (without BCC transmission).	- Data bits sent on line. - Control mode defined in front end working bits. - Burst status Transmit in CSP.	*5D (Note)
Note: FES in diagnostic mode using bit sample facility			

PD16 - FES FRONT END LAYER

This routine tests data management using BSC transmission in normal mode. All tests are done is BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.

ERC	Function	Error Description	RAC
0320	BSC normal mode transmission ETB or ETX character decoded	- Data bits sent on line. - Control mode defined in front end working bits. - Burst status Transmit in CSP.	*5D (Note)
	When ETB or ETX character is decoded by the front end in normal mode, the BCCs or LRC (according to CRC type defined) are send on the line and the front end layer enters the control mode.		
0321	BSC Normal mode transmission DLE-STX sequence decoded	- Transparent mode defined in front end working bits.	*5D (Note)
	When a DLE-STX sequence is decoded in normal mode, the front end layer enters the BSC transparent mode.		
0322	BSC normal mode transmission Test of VCR generation.	- Data bits (with VRC bits) sent on line - Burst status Transmit in CSP	*5D (Note)
	VRC generation (vertical redundancy check) works normally in normal mode. The test is done in: BSC ASCII 7 bits.		
0323	BSC normal mode transmission STX character decoded.	- BCCs value in FES = value of STX character accumulated according to BSC/CRC type.	*5D (Note)
	An STX character decoded in normal mode (in the data flow and not an STX allowing entry to the normal mode) is always accumulated in the BCCs according to CRC type.		
0324	BSC normal mode transmission ITB decoded.	- Data bits sent on line. - Burst status Transmit in CSP. - BCCs or LRC in FES RAM according to CRC type. - Normal mode is defined in front end working bits.	*5D (Note)
	When ITB is decoded in the normal mode: If option is ITB is data, the ITB is processed as a data character. If option is ITB Mode: - No EIB character is to be deleted after ITB. - BCCs are send on the line according to CRC type. - BCCs are preset to the initial value according to CRC type. If option is EIB mode: Same as ITB mode but an (EIB) character is deleted after the ITB character (in the data flow).		
Note: FES in diagnostic mode using bit sample facility			

PD17 - FES FRONT END LAYER

This routine tests data management using BSC transmission in transparent mode, and also tests DLE-SYN generation.

ERC	Function	Error Description	RAC
0325	BSC normal mode transmission Test of DLE-SYN generation	- Data bits sent on line. - Burst Status Transmit in CSP. - level 2 interrupt with underrun condition. - Level 2 interrupt with timeout condition.	*5D (Note)
	- When underrun is detected by FE in BSC normal mode, it generates a DLE-SYN sequence continuously (doing an interrupt level 2 with underrun) until underrun exit. - Every second (timer full) if the front end is in the normal mode and the option SYN Insert is on, the FE generates a DLE-SYN sequence and continues normally. If the option SYN Insert is off, the FES does an interrupt level 2 with a timeout condition. The test is done in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 7 bits.		
0326	BSC transparent mode transmission Test of DLE-SYN generation	- Data bits sent on line. - Burst status Transmit in CSP.	*5D (Note)
	Test if DLE-SYN generation is delayed by the front end. When working bit TE is on (used to send a blocked sequence). The test is done in: BSC EBCDIC.		
Note: FES in diagnostic mode using bit sample facility			

PD18 - FES FRONT END LAYER

This routine tests data management using BSC transmission in transparent mode. All tests are done in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASC II 8 bits.

ERC	Function	Error Description	RAC
0327	BSC transparent mode transmission. DLE-SYN character generation	- BCCs in FES RAM must be equal to an expected value according to CRC type and BSC type.	*5D (Note)
	DLE-SYN characters generated by the front end are not accumulated in the CRC in transparent mode.		
0328	BSC transparent mode transmission. DLE character decoded.	- Data bits sent on line. - BCCs in FES RAM must be equal to an expected value according to CRC and BSC type.	*5D (Note)
	DLE characters decoded by front end with the TE bit off are doubled; but only 1 DLE is accumulated in CRC.		
0329	BSC transparent mode transmission. DLE character decoded.	- Data bits sent on line. - BCCs in FES RAM must be equal to expected value according to CRC and BSC type.	*5D (Note)
	DLE characters decoded by front end with the TE bit on in transparent mode are not doubled and are not accumulated in the CRC.		
0330	BSC transparent mode transmission. All BSC control characters (except DLE)	- Data bits sent on line. - Burst status Transmit in CSP.	*5D (Note)
	All BSC control characters (except DLE) not preceded by a DLE character are processed as data characters in transparent mode.		
0331	BSC transparent mode transmission. Abort procedure.	- Data bits sent on line. - Burst status Transmit in CSP. - Control mode defined in front end working bits.	*5D (Note)
	DLE-ENQ sequence decoded by the FE (with the TE bit on) in transparent mode allows a return to control mode.		
0332	BSC transparent mode transmission. DLE-ETB or DLE-ETX sequence	- Data bits sent on line. - Burst status Transmit in CSP. - Control mode defined in front end working bits.	*5D (Note)
	When a DLE-ETB or DLE-ETX sequence is decoded by the FE (with the TE bit on) in transparent mode, the BCC characters accumulated by the FE are send on the line and the front end returns to the control mode.		
Note: FES in diagnostic mode using bit sample facility			

PD19 - FES FRONT END LAYER

This routine tests data management using BSC transmission in transparent mode, and also tests the CRC/LRC mechanism in BSC transmission.

ERC	Function	Error Description	RAC
0333	BSC transparent mode transmission. VCR generation	- Data bits sent on line. - Burst Status Transmit in CSP. - BCCs in FES RAM must be equal to an expected value according to CRC type	*5D (Note)
	VRC generation does not work in the transparent mode. The tests are done in BSC ASCII 7 bits. All characters are processed as 8 bits characters. CRCs accepted are CRC B and CRC S.		
0334	BSC transparent mode transmission. DLE-ITB sequence decoded	- Data bits sent on line. - Burst Status Transmit in CSP. - BCCs value in FES RAM must be equal to an expected value according to the CRC type. - Transparent mode or normal mode defined in FES RAM according to mode defined.	*5D (Note)
	When DLE-ITB sequence is decoded by front end (with TE bit on) in transparent mode: If option ITB is data: The ITB is processed as a DATA character. If option ITB Mode: - No EIB character is to be deleted after ITB. - BCCs are sent on line according to CRC type. - BCCs are preset to the initial value according to CRC type. - The front end returns to normal mode. If option EIB Mode: Same as ITB mode but an (EIB) character is deleted after ITB (in the data flow).		
0335	BSC transmission. Test of CRC B - CRC S - LRC mechanism.	BCCs or LRC in FES RAM not equal to an expected value defined according to BSC type and CRC type.	*5D (Note)
	3 types of CRC accumulation are possible: CRC B: $X^{16} + X^{15} + X^2 + 1$ CRC S: $X^{16} + X^{12} + X^5 + 1$ LRC : $X^8 + 1$ A transmission of a special pattern allows the complete test of these algorithms. It is done for the different BSC types.		
Note: FES in diagnostic mode using bit sample facility			

PE20 - FES FRONT END LAYER

This routine tests data management using BSC receive. The following functions are tested:

- Synchronization mechanism
- BSC coding receive functions.
- Start function in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	Function	Error Description	RAC
0350	BSC Receive. Test of synchronization research mechanism.	- CP bit (signaling synchro state) in FES RAM does not have the expected value according to the test done. - Data Burst Receive in CSP. - Burst Status Receive in CSP.	*53 (Note)
	When a receive interface is started in BSC, the front end layer is looking for a special pattern (set in RAM B and generally defined with a SYN-SYN value). Before that pattern, the data is not sent to scanner base and CSP. The test is done with synchronization set for 2 characters (of 8 or 7 or 6 bits) or for 1 character (option mono SYN on). This test is done in BSC coding with synchronization found or not. Note: FES in diagnostic mode using bit sample facility		
0351	BSC Coding receive. Normal data management.	- Data burst receive in CSP - Burst Status Receive in CSP.	*5D
	Test of reception done in: BSC coding 8 bits, BSC coding 7 bits, BSC coding 6 bits Note: FES in diagnostic mode using bit injection facility		
0355	BSC Receive. Start Processing.	- Synchronization research state must be defined in front end working bits	*53
	Start function on a specific interface; the corresponding interface must start in the synchronization research state. The test is done in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits. Note: FES in diagnostic mode		
0356	BSC Receive. Test of synchronization mechanism.	- CP Bit in FES RAM must be found on - Data Burst Receive in CSP - Burst Status Receive in CSP.	*53
	The test is done in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits. Note: FES in diagnostic mode using bit injection facility		

PE21 - FES FRONT END LAYER

This routine tests data management using BSC receive in control mode. All tests are done in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	Function	Error Description	RAC
0357	BSC control mode receive. SYN character deletion	- Data Burst Receive in CSP - Burst Status Receive in CSP.	*5D (Note)
	When SYN character is found in the data flow, in control mode, it is deleted.		
0358	BSC control mode receive. SYN-SYN sequence and action on timer.	- FES working bit e4 (timer force 30) must be found on or off according to the test.	*53 (Note)
	When a SYN-SYN sequence is detected by front end, the front end timer, tracking the loss of synchronization, is reinitialized for 3 seconds. On a continuous SYN-SYN sequence, however, the timer is activated only on the first SYN-SYN.		
0359	BSC control mode receive. Data character after a SYN-SYN sequence and action on timer.	- FES working bit e4 (timer force 30) must be found on or off according to the test done.	*53 (Note)
	When a data character is detected by front end after a SYN-SYN sequence, the front end timer, tracking the loss of synchronization, is reinitialized for 3 seconds. On a continuous data sequence, however, the the timer is activated only on the first data decode.		
0360	BSC control mode receive. ITB-ETB-ETX character decode	- Data Burst Receive in CSP - Burst Status Receive in CSP.	*5D (Note)
	ITB - ETB - ETX characters decoded in control mode are processed in the same way as data characters.		
Note: FES in diagnostic mode using bit injection facility			

PE22 - FES FRONT END LAYER

This routine tests data management using BSC receive in control mode. All tests are done in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	Function	Error Description	RAC
0361	BSC control mode receive. EOT, NAK, ENQ, ACK0, ACK1, WACK, RVI character decode	- Data Burst Receive in CSP - Burst Status Receive in CSP. - FES working bit defined monitoring for PAD State	*5D
	EOT, NAK, ENQ, ACK0, ACK1, WACK, and RVI characters decoded in the control mode cause a timer force 0 and force the FES to enter the Monitoring for PAD state.		
Note: FES in diagnostic mode using bit injection facility			

PE23 - FES FRONT END LAYER

This routine tests data management using BSC receive in control mode.

ERC	Function	Error Description	RAC
0362	BSC control mode receive. STX or SOH character decoded	- FES working bits do not define 'enter normal'. - FES working bit does not define FES in normal mode - FES working bit e4 (timer force 30) must be on. - Interrupt level 2 (enter normal) not done to CSP.	x53 (Note)
	STX or SOH character decoded in control mode force the front end layer to enter the normal mode: - Cause an interrupt level 2 to the CSP. - Reinitialize the 3-second timer. The tests are done with STX character, then with SOH character. The test is done in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.		
0363	BSC control mode receive. DLE-STX or DLE-SOH sequence	- FES working bits do not define enter transparent. - FES working bit does not define FES FES in transparent mode - FES working bit e4 (timer force 30) force 30) must be on. - Interrupt level 2 (enter transparent) not done to CSP.	x53 (Note)
	A DLE-STX or DLE-SOH sequence decoded in control mode forces the front end layer to enter into the transparent mode: - Cause an interrupt level 2 to the CSP. - Reinitialize the 3-second timer. The tests are done with DLE- STX sequence, then with DLE- SOH sequence. The test is done in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.		
0364	BSC control mode receive. VCR checking	- VRC check is not reported in burst status receive in CSP	x5F (Note)
	VRC checking works normally in control mode for data characters and BSC control characters. Test that a bad VRC is detected and immediately reported as a a VRC check in ASCII 7 bits for data and control characters.		
0365	BSC control mode receive. VCR deletion	- Data Burst Receive in CSP (must have VRC deleted).	x5D (Note)
	VRC Deletion works normally in control mode for data characters and BSC control characters. The test done in ASCII 7 bits for data and control characters.		
0366	BSC control mode receive. Test of overrun processing	- FES working bit 'overrun' must be on. - FES working bit 'data check remembrance' must be off. - Interrupt level 2 with overrun condition must be done to CSP.	x58 (Note)
	Test of overrun processing in control mode. When an overrun condition is detected in control mode:. - FES raises an interrupt L2 with overrun condition. - However, no data check remembrance is done in control mode.		
Note: FES in diagnostic mode using bit injection facility			

PE24 - FES FRONT END LAYER

This routine tests data management using BSC receive normal mode. All tests are done in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bit s.

ERC	Function	Error Description	RAC
0367	BSC normal mode receive. SYN character deletion.	- Data Burst Receive in CSP. - Burst Status Receive in CSP.	*5D (Note)
	When a SYN character is found in the data flow in normal mode, it is deleted.		
0368	BSC normal mode receive. SYN-SYN sequence and action on timer.	- FES working bit E4 (timer force 30) must be found on or off according to the test done.	*53 (Note)
	When a SYN-SYN sequence is detected by the front end in normal mode, the front end timer, tracking the loss of synchronization, is reinitialized for 3 seconds. On a continuous SYN-SYN sequence, the timer is activated only on the first SYN-SYN.		
0369	BSC normal mode receive. Data character after SYN-SYN sequence and action on timer.	- FES working bit E4 (timer force 30) must be on or off according to the test done.	*53 (Note)
	When a data character is detected by front end after a SYN-SYN sequence in normal mode, the front end timer, tracking the loss of synchronization, is reinitialized for 3 seconds. On a continuous data sequence the timer is activated only on the first first data decoded.		
0370	BSC normal mode receive. SYN character and action on CRC accumulation.	- FES RAM for BCC fields must be equal to an expected value according to BSC/SCR type.	*53 (Note)
	SYN characters are not accumulated in the CRC in the normal mode.		
0371	BSC normal mode receive. ENQ processing.	- Data Burst Receive in CSP. - Burst status receive in CSP. - FES working bit must define monitoring for PAD state	*5D (Note)
	When an ENQ character is decoded in the normal mode: - A burst change is forced with an interrupt to CSP. - The FES enters into the monitoring for PAD state.		
Note: FES in diagnostic mode using bit injection facility			

PE25 - FES FRONT END LAYER

This routine tests data management using BSC receive in normal mode. The routine also tests CRC B, CRC S, and LRC accumulation, and the ETB-ETX process.

ERC	Function	Error Description	RAC
0372	BSC normal mode receive. Test of CRC B - CRC S - LRC mechanism.	- Data Burst receive in CSP. - Burst Status receive in CSP. - FES working bit must define front end in synchronization research state.	*5D (Note)
	3 types of CRC accumulation are possible: CRC B: $X^{16} + X^{15} + X^2 + 1$ CRC S: $X^{16} + X^{12} + X^5 + 1$ LRC : $X^8 + 1$ The reception of a special pattern allows the complete test of algorithms. It is done for the different BSC types. When ETB or ETX is detected by front end, the characters following are the BCCs (or LRC). The BCCs accumulated by the front end must be equal to these characters If yes: burst change with CRC OK. If no : burst change with CRC not OK. After BCC phase the front end must be in synchronization research. Different tests are done for all BSC/CRC types with CRC OK and with CRC not OK on each bit of BCC characters.		
Note: FES in diagnostic mode using bit injection facility			

PE26 - FES FRONT END LAYER

This routine tests data management using BSC receive in normal mode.

It also tests ITB processing. All tests are done in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	Function	Error Description	RAC
0373	BSC normal mode receive. Test of ITB Processing.	- Data Burst receive in CSP. - Burst Status receive in CSP.	*5D (Note)
	When ITB is decoded in normal mode: If ITB is data, the ITB is processed as a data character. If ITB mode, there is no EIB to generate after ITB. If CRC OK, nothing happens. If CRC not OK, data check remembrance is set on and is reported at the at the next ETB or ETX encountered. The front end continues in normal mode. If EIB mode, there is an EIB character to generate after the ITB character. If CRC OK, nothing happens except EIB generation. If CRC not OK, it is immediately reported (with EIB character) with special status. The front end continues in normal mode. If EIB mode with burst change there is an EIB character. If CRC OK, after EIB generation a special status is reported to CSP. If CRC not OK after EIB generation another special status is reported to CSP.		
Note: FES in diagnostic mode using bit injection facility			

PE27 - FES FRONT END LAYER

This routine tests data management using BSC receive in normal mode.

ERC	Function	Error Description	RAC
0374	BSC normal mode receive. Test of DLE-STX sequence decoded in normal mode without 'EIB Mode + Burst Change' option:	- Burst Status receive in CSP. - FES working bit must define FES in transparent mode	*53 (Note)
	Force the front end to enter the transparent mode, but without reporting it to the CSP (by status). The test is done in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.		
0375	BSC normal mode receive. Test of DLE-STX sequence decoded in normal mode with 'EIB Mode + Burst Change' option.	- FES working bit must define FES in transparent mode. - FES working bit must signal enter transparent and timer force 30 must be on.	*53 (Note)
	Force the front end to enter the transparent mode and report it to the CSP (by a status) and the 3-second timer is restarted. The test is done in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.		
0376	BSC normal mode receive. Test of VRC Checking mechanism.	- Burst status receive in CSP:	*5F
	In normal mode, a bad VRC is detected but its reporting, as a VRC check, is delayed until the first BCC phase encountered. The test is done in ASCII 7 bits for a bad VRC on data characters and on BSC control characters.		
0377	BSC normal mode receive. Test of VRC Deletion.	- Data Burst receive in CSP. - Burst Status receive in CSP.	*5D (Note)
	The VRC bit is deleted in normal mode for data characters and BSC control characters. The test is done in ASCII 7 bits.		
0378	BSC normal mode receive. EOT and NAK processing	- Data Burst receive in CSP. - Burst Status receive in CSP.	*5D (Note)
	EOT and NAK character are processed as data characters in normal mode. The test is done in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.		
0379	BSC normal mode receive. Overrun processing.	- Interrupt level 2 with overrun condition must be sent to CSP. - FES working bits must signal overrun and data check.	*5A (Note)
	When overrun condition is detected by front end in normal mode: - An interrupt level 2 is sent to CSP with overrun condition. - Data check remembrance is saved and the data check reporting is delayed until the first BCC phase is encountered.		
Note: FES in diagnostic mode using bit injection facility			

PE28 - FES FRONT END LAYER

This routine tests data management using BSC receive in transparent mode. All tests are done in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	Function	Error Description	RAC
0380	BSC transparent mode receive Test of DLE-SYN Deletion.	- Data Burst receive in CSP. - Burst Status receive in CSP.	*5D (Note)
	When a DLE-SYN sequence is found in the data flow in transparent mode, it is deleted.		
0381	BSC transparent mode receive Test ot DLE-SYN sequence and action on CRC accumulation.	- FES RAM for BCC fields must be equal to an expected value according to BSC/CRC type.	*53 (Note)
	The DLE-SYN sequence is not accumulated in BCCs/LRC in the transparent mode.		
0382	BSC transparent mode receive DLE-SYN sequence and action on timer.	- FES working bit e4 (timer force 30) must be found on or off according to the test done.	*53 (Note)
	When a DLE-SYN sequence is detected by the front end in transparent mode, the front end timer, tracking the loss of synchronization is reinitialized for 3 seconds. On a continuous DLE-SYN sequence the timer is activated only on first DLE-SYN.		
0383	BSC transparent mode receive Data character after DLE-SYN sequence and action on timer	- FES working bit e4 (timer force 30) must be found on or off according to the test done.	*53 (Note)
	When a data character is detected by front end after a DLE-SYN sequence in transparent mode, the front end timer is reinitialized for 3 seconds. On a continuous data sequence the timer is activated only on the first data decode.		

(PE28, continued)

ERC	Function	Error Description	RAC
0384	BSC transparent mode receive DLE-DLE sequence processing.	- Data Burst receive in CSP. - Burst Status receive in CSP. - BCCs field in FES RAM must be equal to an expected value according to to BSC/CRC type.	*5D (Note)
	When a DLE-DLE sequence is decoded by the front end in transparent mode, one DLE is deleted from the data flow. It is not accumulated in the BCCs. The other DLE is a data character, it is sent to the scanner base and CSP and it is accumulated in the BCCs.		
0385	BSC transparent mode receive Test of Invalid DLE Sequence	- Data Burst receive in CSP. - Burst Status receive in CSP. - FES working bit must define the front end in transparent mode	*5D (Note)
	In transparent mode, after a DLE character, any SOH, EOT, NAK, or data character, is signaled as an invalid DLE sequence (in status) and the front end continues in transparent mode (only DLE, SYN, ITB, ETB, ETX, ENQ, and STX are a valid sequence).		
0386	BSC transparent mode receive DLE-ENQ sequence processing.	- Data Burst receive in CSP. - Burst Status receive in CSP. - FES working bit must define the front end in monitoring for PAD state.	*5D (Note)
	A DLE-ENQ sequence detected by front end in transparent mode is signaled in status and force the front end to enter the monitoring for PAD state.		
Note: FES in diagnostic mode using bit injection facility			

PE29 - FES FRONT END LAYER

This routine tests data management using BSC receive in transparent mode. All tests are done in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	Function	Error Description	RAC
0387	BSC transparent mode receive DLE-ETB or DLE-ETX processing.	- Data Burst receive in CSP. - Burst Status receive in CSP.	*5D (Note)
	When a DLE-ETB or a DLE-ETX sequence is detected by the front end in transparent mode, the front end signals the sequence to the CSP (in status) after a BCC phase checking. The result of the BCC checking is given in the next burst with BCCs OK or BCCs not OK in the status.		
0388	BSC transparent mode receive DLE-ITB sequence processing.	- Data Burst receive in CSP. - Burst Status receive in CSP. - FES working bit does not signal front end in the expected BSC mode depending on the test done.	*5D (Note)
	When a DLE-ITB sequence is decoded by the front end in transparent mode If ITB is Data: It is signaled in status as an invalid DLE sequence and the front end continues in the transparent mode. If ITB Mode: There is no EIB character to be generated after the ITB character. If CRC OK: no action. If CRC not OK: data check remembrance is set on and is reported at the next ETB or ETX encountered. The front end enters the normal mode. If EIB Mode: There is an EIB character to be generated after the ITB character. If CRC OK: no action except EIB generation. If CRC not OK: it is immediately reported (with EIB character) with aspecial status. The front end enters the normal mode. If EIB mode with Burst Change: There is an EIB character to be generated after the ITB character. If CRC OK: after EIB generation a special status is reported to CSP If CRC not OK: after EIB generation another special status is reported to the CSP. The front end layer enters the normal mode.		
Note: FES in diagnostic mode using bit injection facility			

PE30 - FES FRONT END LAYER

This routine tests data management using BSC receive in transparent mode.

ERC	Function	Error Description	RAC
0389	BSC transparent mode receive BSC control characters without DLE.	- Data Burst receive in CSP. - Burst Status receive in CSP. - FES working bit must define the front end in transparent mode.	*5D
	All BSC control characters decoded by the front end in transparent mode without a preceding DLE are processed as data characters. The test is done in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits, for following characters: STX-ETX-SOH-SYN-ETB- ITB-ENQ-EOT-NAK-ACK0 ACK1-WACK-RVI.		
0390	BSC transparent mode receive VRC checking.	- Burst Status receive in CSP.	*5D (Note)
	VRC checking function does not work in transparent mode Test done in ASCII 7 bits.		
0391	BSC transparent mode receive VRC deletion.	- Data Burst receive in CSP.	*5D (Note)
	VRC deletion function does not work in transparent mode Test done in ASCII 7 bits.		
0392	BSC transparent mode receive Overrun processing.	- FES working bit must signal overrun and data check - Interrupt level 2 with overrun condition must be done to CSP. done to CSP.	*58 (Note)
	When overrun condition is detected by the front end in transparent mode: - An interrupt level 2 is sent to CSP with overrun condition. - Data check remembrance is saved and data check reporting is delayed until the first BCC phase encountered (DLE-ITB - DLE-ETB - DLE-ETX).		
Note: FES in diagnostic mode using bit injection facility			

PE31 - FES FRONT END LAYER

This routine tests data management using BSC receive, and also monitors for PAD processing.

ERC	Function	Error Description	RAC
0393	BSC receive. Monitoring for PAD processing.	- Data Burst receive in CSP. - Burst Status receive in CSP. - FES working bit must define front end in synchronization research state	*5D (Note)
	In this state the next character received must have the 4 first bits set to 1. If PAD OK: status given to CSP with pad OK. If PAD not OK: status given to CSP with pad not OK. After PAD checking the front end enters the synchronization research status.		
Note: FES in diagnostic mode using bit injection facility			

PE32 - FES/ALC

This routine tests the receive interface of the ALC feature. The Transmit interface, similar to BSC coding, being already covered, this routine tests the detection and reporting of ALC specific bit patterns.

ERC	Function	Error Description	RAC
0400	Test synchronization pattern Character assembling (6-bit length). Test SIN1 remember bit.	ALCSYN is not recognized. Bad pattern received. SYNC1 remember bit in RAM C off.	x52
0401	Delete SYN1 character when SYN1 remember bit on. Reset SIN1 remember bit. SYNPAD (SYN1 SYN2) detection and reporting (ending flag 2 interrupt).	- SYN1 character is not deleted from received pattern - SYNC1 remember bit stays on in RAM C Bad status is detected.	
0402	Changing of the burst at each SYNPAD detection.	Data in bursts are not correct	
0403	SYNPAD detection and reporting out off character boundary with the restart on a new boundary.	Bad data burst is received. Bad status is detected.	

PF60 - FES SYNCHRONOUS OPERATION

This routine tests the service bus between the FES and the LICs and the corresponding drivers and receivers.

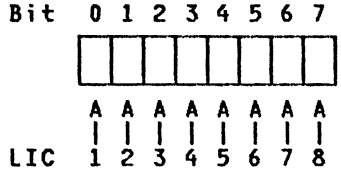
ERC	Function	Error Description	RAC
0701	Test of service bus between FES and LICs.	If only one LIC is found in error If more than one LIC is found in error, then: (ERRBIT: flag the LICs found in error).	x86 x65
	One character is wrapped between transmit and receive interfaces (the wrap is done in the LIC used in the wrap mode). The test exercises all LIC cards installed, and the first line installed on each LIC (except for LIC 1 where the first non-autocall line is used). The test verifies if data character is correctly wrapped.		
Note: LICs are used in diagnostic mode using LIC internal data wrap facility			

PF61 - FES SYNCHRONOUS OPERATION

This routine tests the bus for the transmit bit (before and after the LIC drivers) and the corresponding drivers/receivers.

ERC	Function	Error Description	RAC
0702	Test of bus for transmit bit (before/after LIC drivers.	If only one LIC is found in error.	*86
		If more than one LIC is found in error, then: (ERRBIT: flag the LICs found in error).	*65
	One character = X '55' (mark space...) is transmitted for about 2 seconds with function 'driver check on bit mark' unmasked on corresponding line, in order to track a possible driver check on bit mark. The test exercises all LIC cards installed, and the first line installed on each LIC (except for LIC 1 where the first non-autocall line is used). The test verifies if there is no driver check on bit mark.		
Note: Local attachment clock is used for the transmission			

In this routine, the first 8 bits of the ERR BIT field on the screen indicate which card is failing:



The LIC card will only be tested if it has a line defined in the 3725 CDF.

QA01 - LIC CARD TEST: CARD IDENTIFICATION REGISTER TEST

This routine checks that the LIC card under test has the correct type identification.

FUNCTION: Card identification register (reg 11) for first line of LIC card under test is read and its contents is checked against LIC identification information given by CDF.

ERC	RAC	Error Description
0611	*84	Bits 0, 1, 2, and 3 of LIC register 11 do not contain the expected value.

QA02 - LIC CARD TEST: ADDRESS BUS PARITY CHECKER TEST

This routine checks that the address bus parity checker of the LIC card under test is error free. Only the line configuration in use is tested in this routine.

FUNCTION: All LIC registers (00-01-10-11) for all lines installed on the LIC card under test are addressed in read mode with a good parity and a bad parity.

STEP:

1. If parity is OK: Read operation is complete (result given in FES XR16).
2. If parity is not OK: Read operation is not complete (result given in FES XR16).

ERC	RAC	Error Description
0611	*86	Read operation not complete (with parity OK, Step 1).
0612	*86	Read operation complete (with parity not OK, Step 2).

QA03 - LIC CARD TEST FOR LIC1 - LIC2 - LIC3: LINE SELECTION TEST

This routine ensures that the line selection mechanism of the LIC1, 2, or 3 card under test is working correctly. Only the line configuration in use is tested in this routine.

FUNCTION: LIC register 01 (Modem-Out) for lines 0 through 3 is written (bits 0 through 3) with the line address value it belongs to. Reading each register back from line 0 to line 3 allows comparison between written pattern and read pattern. The same operations are then performed again, but from line 3 to line 0.

ERC	RAC	Error Description
0611	*83	LIC Modem-Out register (01) does not contain expected value (for bits 0 to 3).

QA04 - LIC CARD TEST (FOR LIC4): LINE SELECTION TEST

Same as routine QA03 but for LIC4.

Note: A clock time must be sent before reading back Modem-Out data from LIC4 card (this is the difference from the previous routine).

ERC	RAC	Error Description
0611	*83	LIC Modem-Out register (01) does not contain expected value (for bits 0 to 3).

QA05 - LIC CARD TEST (LINE CONFIGURATION REG): GENERAL RESET TEST

This routine ensures that the general reset lead coming from the FES card is working correctly in the LIC card under test. It is assumed that only the 'line configuration latches' (clock mode) are to be tested in this routine.

FUNCTION: Each 'line configuration register' of the LIC card under test is written with 1s (clock mode = 11). After verifying that each register is effectively set on, the 'general reset' lead is activated and each register content is checked to be X'0' (for clock information).

STEP:

1. Pretest phase: Write of '11' to 'line configuration register' not effective.

2. Test Phase: General reset not effective for 'line configuration register'.

ERC	RAC	Error Description
0411	*83	Value of 'clock mode' bits not equal to '11' (Step 1, test phase)
0611	*83	Value of 'clock mode' bits not equal to '00' (Step 2, pretest phase)

QA06 - LIC CARD TEST (LIC 1 MODEM-OUT REGISTER): GENERAL RESET TEST

This routine ensures that the general reset lead coming from the FES card is working correctly in the LIC 1 card under test. It is assumed that only the 'Modem-Out latches' are to be tested in this routine (V24 and V25 interfaces).

FUNCTION: Each 'line Modem-Out register' of the LIC 1 card under test is written with X'FC' for V25 or X'78' for V24. After verifying that each register is effectively set on, the 'general reset' lead is activated and each register content is checked to be X'0'.

STEP:

1. Pretest phase: Write Modem-Out register not effective.

2. Test Phase: General reset not effective for 'Modem-Out register'.

ERC	RAC	Error Description
0411	*8A	Pretest phase (Step 1): - Value of 'Modem-Out register not equal to X'FC' if V25 - Value of 'Modem-Out register not equal to X'78' if V24 Note: can be due to a mismatch between V24 and V25 cable in CDF
0611	*85	Test Phase (Step 2): - Value of 'Modem-Out register not equal to X'00'

QA07 - LIC CARD TEST (LIC 2 MODEM-OUT REGISTER): GENERAL RESET TEST

This routine ensures that the general reset lead coming from the FES card is working correctly in the LIC 2 card under test. It is assumed that only the 'Modem-Out latches' are to be tested in this routine.

FUNCTION: Each 'line Modem-Out register' of the LIC 2 card under test is written with X'C8'. After verifying that each register is effectively set on, the 'general reset' lead is activated and each register content is checked to be X'0'.

STEP:

1. Pretest phase: Write Modem-Out register not effective.
2. Test Phase: General reset not effective for 'Modem-Out register'.

ERC	RAC	Error Description
0411	*83	Pretest phase (Step 1): Value of 'Modem-Out register not equal to X'C8'
0611	*85	Test Phase (Step 2): Value of 'Modem-Out register not equal to X'00'

QA08 - LIC CARD TEST (LIC 3 MODEM-OUT REGISTER): GENERAL RESET TEST

This routine ensures that the general reset lead coming from the FES card is working correctly in the LIC 3 card under test. It is assumed that only the 'Modem-Out latches' are to be tested in this routine.

FUNCTION: Each 'line Modem-Out register' of the LIC 3 card under test is written with X'E8'. After verifying that each register is effectively set on, the 'general reset' lead is activated and each register content is checked to be X'0'.

STEP:

1. Pretest phase: Write Modem-Out register not effective.
2. Test Phase: General reset not effective for 'Modem-Out register'.

ERC	RAC	Error Description
0411	*83	Pretest phase (Step 1): Value of 'Modem-Out register not equal to X'F8'
0611	*83	Test Phase (Step 2): Value of 'Modem-Out register not equal to X'00'

QA09 - LIC CARD TEST (LIC 4 MODEM-OUT REGISTER): GENERAL RESET TEST

This routine ensures that the general reset lead coming from the FES card is working correctly in the LIC 4 card under test. It is assumed that only the 'Modem-Out latches' are to be tested in this routine.

FUNCTION: Each 'line Modem-Out register' of the LIC 4 card under test is written with X'E4'. After verifying that each register is effectively set on, the 'general reset' lead is activated and each register content is checked to be X'0'.

STEP:

1. Pretest phase: Write Modem-Out register not effective.
2. Test Phase: General reset not effective for 'Modem-Out register'.

ERC	RAC	Error Description
0411	*83	Pretest phase (Step 1): Value of 'Modem-Out register not equal to X'E0'
0611	*83	Test Phase (Step 2): Value of 'Modem-Out register not equal to X'00'

QB01 - LIC LINE TEST: CABLE IDENTIFICATION REGISTER TEST.

This routine checks that the cable connected to the line under test matches the relevant value fetched from the CDF.

FUNCTION: The 'Line Cable identification register' (reg 10), is read and its contents (for bits 3, 4, and 5 is compared against the corresponding CDF value.

ERC	RAC	Error Description
0611	*85	Bits 3, 4, 5 of LIC register 10 do not contain expected value.

QB02 - LIC LINE TEST (LIC1 - LIC2 - LIC3): LINE RESET FUNCTION TEST

This routine ensures that the selective 'line reset function' is working correctly for LIC 1, LIC 2, and LIC 3 lines. It is assumed that only the 'Modem-Out latches' are to be tested in this routine.

FUNCTION: Each 'Modem-Out register' of the line under test is written with X'C0'. After verifying that the register is effectively set on, the 'line reset function' is activated and register contents is checked to be X'0'.

STEP:

1. Pretest phase: Write Modem-Out register not effective.
2. Test Phase: Line reset function not effective for 'Modem-Out register'.

ERC	RAC	Error Description
0411	*83	Pretest phase (Step 1): Value of 'Modem-Out register not equal to X'C0'
0611	*83	Test Phase (Step 2): Value of 'Modem-Out register not equal to X'00'

QB03 - LIC LINE TEST (LIC 4): LINE RESET FUNCTION TEST

This routine ensures that the selective 'line reset function' is working correctly for LIC 4 lines. It is assumed that only the 'Modem-Out latches' are to be tested in this routine.

FUNCTION: Each 'Modem-Out register' of the line under test is written with X'C4'. After verifying that the register is effectively set on, the 'line reset function' is activated and register contents is checked to be X'0'.

STEP:

1. Pretest phase: Write Modem-Out register not effective.
2. Test Phase: Line reset function not effective for 'Modem-Out register'.

ERC	RAC	Error Description
0411	*83	Pretest phase (Step 1): Value of 'Modem-Out register not equal to X'C0'
0611	*83	Test Phase (Step 2): Value of 'Modem-Out register not equal to X'00'

QB04 - LIC LINE TEST: LINE CONFIGURATION REGISTER TEST

This routine checks that the latches of the clocking mode function are working correctly for the line under test.

FUNCTION: The four combinations (00 - 01 - 10 - 11) of the two clock mode bits are written into clocking mode register (bit 4 and 5 of line configuration register). After each write operation, a read operation is performed and register content is checked up for clock mode bits.

ERC	RAC	Error Description
0611	*83	LIC Line configuration register does not contain expected value for bits 4 and 5

QB05 - LIC LINE TEST: DATA BIT TRANSFER TEST

This routine checks that the data bit handling performed by the LIC line works correctly. For the autocall line, the routine is selected but the tests are not done.

FUNCTION: A bit string, initiated in the FES card, is serialized out to the line under test on the LIC card. By using the diagnostic clock (in the LIC) and the Modem-Out wrap register, the bit transfer from the FES card to the LIC driver is checked. The transmitted bit is transferred back to the FES card by using the LIC wrap capability and the bit string is checked.

STEP:

1. Pretest phase: Transmit path in error.

2. Test phase: Receive path in error.

ERC	RAC	Error Description
0411	*83	Pretest phase (Step 1). Data bit to transmit not equal to transmit bit echo read in LIC Modem-Out wrap register.
0611	*86	Test phase (Step 2). Data bit string received in FES RAM B is not equal to the expected string.

QB06 - LIC LINE TEST: DATA BIT TRANSFER - LINE RESET FUNCTION TEST

This routine checks that the latches used to handle the data bit transfer for each line of the LIC card are correctly reset by the line reset function. For the autocall line, the routine is selected but the tests are not done.

FUNCTION: A bit string, initiated in the FES card, is serialized out to the line under test. The diagnostic clock is activated twice and then the 'line reset function' is activated. The rest of the pattern is sent back to the FES card by the LIC internal wrap capability. A check is made in the FES card between the sent and received patterns.

ERC	RAC	Error Description
0611	*83	LIC 1/2/3 in error. Received pattern is not the expected one (X'0C').
0612	*83	LIC 4 in error. Received pattern is not the expected one (X'00').

QC01 - LIC LINE TEST, CE WRAP INSTALLED: LIC 1 MODEM-IN REGISTER TEST

This routine ensures that the receivers of the Modem-In register for each LIC 1 line are working correctly.

FUNCTION: When a CE wrap block is connected on the tailgate, LIC Modem-Out signals are wrapped with LIC Modem-In signals.

In order to test the line receivers of LIC Modem-In register, 2 patterns are sent and checked (X'D8' and X'00').

ERC	RAC	Error Description
0611	*8B	LIC Modem-In register does not contain the expected value (X'F8' and X'00').

QC02 - LIC LINE TEST, CE WRAP INSTALLED: LIC 2 MODEM-IN REGISTER TEST

This routine ensures that the receivers of the Modem-In register for each LIC 2 line are working correctly.

FUNCTION: When a CE wrap block is connected on the tailgate, LIC Modem-Out signals are wrapped with LIC Modem-In signals.

In order to test the line receivers of LIC Modem-In register, 2 patterns are sent and checked (X'C0' and X'00').

ERC	RAC	Error Description
0611	*8B	LIC Modem-In register does not contain the expected value (X'F0' and X'00').

QC03 - LIC LINE TEST, CE WRAP INSTALLED: LIC 3 MODEM-IN REGISTER TEST

This routine ensures that the receivers of the Modem-In register for each LIC 3 line are working correctly.

FUNCTION: When a CE wrap block is connected on the tailgate, LIC Modem-Out signals are wrapped with LIC Modem-In signals.

In order to test the line receivers of LIC Modem-In register, 2 patterns are sent and checked (X'C8' and X'00').

ERC	RAC	Error Description
0611	*8B	LIC Modem-In register does not contain the expected value (X'D8' and X'00').

QC04 - LIC LINE TEST, CE WRAP INSTALLED: LIC 4 MODEM-IN REGISTER TEST

This routine ensures that the receivers of the Modem-In register for each LIC 4 line are working correctly.

FUNCTION: When a CE wrap block is connected on the tailgate, LIC Modem-Out signals are wrapped with LIC Modem-In signals.

In order to test the line receivers of LIC Modem-In register, 2 patterns are sent and checked (X'C4' and X'00').

ERC	RAC	Error Description
0611	*8B	LIC Modem-In register does not contain the expected value (X'40' and X'00').

QC05 - LIC LINE TEST, CE WRAP INSTALLED: LIC 1 AND 4 DATA RECEIVER TEST

This routine checks that the receiver used to handle the data transfer is working correctly in each LIC 1 and LIC 4 line.

For the autocall line, the routine is selected, but the tests are not done for LIC1.

FUNCTION: A CE wrap block is connected at the tailgate. A pattern, initiated in the FES card, is serialized out to the line under test. The 'New Sync latch' is used as a clock to transfer the data. The transmitted bit is transferred back to the FES card using the wrap block. By reading the receive bit, check that the receive data receiver and the clock receivers are working correctly.

STEP:

1. Bit-by-bit verification.

2. Bit string verification.

ERC	RAC	Error Description
0511	*8B	Bit-by-bit verification (Step 1). Data bit transmitted (bit echo, read in Modem-Out wrap register) is not equal to receive bit (read in Modem- In register).
0611	*86	Bit string verification (Step 2). Data bit string received in FES RAM B is not equal to the expected string.

QC06 - LIC LINE TEST, CE WRAP INSTALLED - LIC 2 DATA RECEIVER TEST

This routine checks that the receiver used to handle the data transfer is working correctly in each LIC 2 line.

FUNCTION: A CE wrap block is connected at the tailgate. A pattern, initiated in the FES card, is serialized out to the line under test. The 'Test latch' is used as a clock to transfer the data. The transmitted bit is transferred back to the FES card using the wrap block. By reading the receive bit, check that the receive data receiver and the clock receivers are working correctly.

STEP:

1. Bit-by-bit verification.
2. Bit string verification.

ERC	RAC	Error Description
0511	*8B	Bit-by-bit verification (Step 1). Data bit transmitted (bit echo, read in Modem-Out wrap register) is not equal to receive bit (read in Modem- In register).
0611	*86	Bit string verification (Step 2). Data bit string received in FES RAM B is not equal to the expected string.

QC07 - LIC LINE TEST, CE WRAP INSTALLED - LIC 3 DATA RECEIVER TEST

This routine checks that the receiver used to handle the data transfer is working correctly in each LIC 3 line.

FUNCTION: A CE wrap cable is connected at the tailgate. A pattern, initiated in the FES card, is serialized out to the line under test. The 'New Sync latch' is used as a clock to transfer the data. The transmitted bit is transferred back to the FES card using the wrap cable. By reading the receive bit, check that the receive data receiver and the clock receivers are working correctly.

STEP:

1. Bit-by-bit verification.
2. Bit string verification.

ERC	RAC	Error Description
0511	*8B	Bit-by-bit verification (Step 1). Data bit transmitted (bit echo, read in Modem-Out wrap register) is not equal to receive bit (read in Modem- In register).
0611	*86	Bit string verification (Step 2). Data bit string received in FES RAM B is not equal to the expected string.

QD01 - LIC LINE TEST - MANUAL INTERVENTION ROUTINE FOR JAPAN

(LIC 1, LIC 3, and LIC 4 NTT On/Off Driver Test)

This routine sets permanently 'on' or 'off' all the drivers used by a line on a LIC 1, 3, or 4 card, so as to allow measurement by the NTT maintenance personnel. This test takes place at installation time according to NTT requirements.

FUNCTION: The CE is requested to choose a state 'on' or 'off' for the drivers of a chosen line. Following the answer, the drivers are set to required state and a message is sent to the MOSS operator console to indicate that the drivers ready for measuring. Measurements are made at the end of the cable coming from the 3725 to the modem. **Error description:** If the measurement does not give the expected result, according to the chosen state, and all the previous TSS diagnostic routines are OK, the cable between 3725 and modem must be changed.

QD02 - LIC LINE TEST - MANUAL INTERVENTION ROUTINE FOR JAPAN

(LIC 1, LIC 3, and LIC 4 NTT Data Wrap Test)

This routine is used to perform a data wrap test using the NTT wrap block located at the external cable end. For the autocall line, the routine is selected but the tests are not done. The test takes place at installation time at the request of the NTT maintenance personnel.

FUNCTION: Only data receivers are in the scope of this routine. It must be pointed out that with NTT wrap, the transmit clock cannot be wrapped to the receive clock, so these signals cannot be verified for external cables in Japan.

A data pattern, initiated in FES card, is serialized out to the line under test.

The LIC diagnostic bit strobe clock is used to transfer the data.

The transmitted bit is transferred back to the LIC card by using the NTT wrap block.

Note: Before the routine selection, transmit and receiver data signals have to be wrapped together on the NTT wrap block.

ERC	RAC	Error Description
0611	*A0	By comparing the transmit bit (read in LIC Modem-In wrap register) with the receive bit (read in LIC Modem-In register) it may be verified that the cable is working correctly.

QD03 - LIC LINE TEST - MANUAL INTERVENTION ROUTINE FOR JAPAN

(LIC 1 NTT Modem-In Wrap Test)

This routine is used to perform a wrap test on the modem control leads using the NTT wrap block located at the external cable end for a LIC 1 line (V24 and V25 interface). The test takes place at installation time at the request of the NTT maintenance personnel.

FUNCTION: Write a pattern into the LIC Modem-Out register of the line and read the contents of the wrapped pattern into LIC Modem-In register of the same line. Compare the Modem-In register contents with the expected pattern.

STEP:

1. V24 interface tests:

   Modem-Out values: X'C0' to X'00'.
   Modem-In expected values: X'D0' to X'00'.

2. V25 interface tests:

   Modem-Out values: X'FC' to X'00'.
   Modem-In expected values: X'A0' to X'5C'.

Note: Before the routine selection, all the signals must be wrapped together according to NTT specifications.

ERC	RAC	Error Description
0611	*A0	Modem-In does not contain the expected value (V24, Step 1).
0612	*A0	Modem-In does not contain the expected value (V25, Step 2).

QD04 - LIC LINE TEST - MANUAL INTERVENTION ROUTINE FOR JAPAN

(LIC 3 NTT Modem-In Wrap Test)

This routine is used to perform a wrap test on the modem control leads using the NTT wrap block located at the external cable end for a LIC 3 line. The test takes place at installation time at the request of the NTT maintenance personnel.

FUNCTION: Write a pattern into the LIC Modem-Out register of the line and read the contents of the wrapped pattern into the LIC Modem-In register of the same line.

Then compare the Modem-In register content to the expected pattern. 2 patterns are sent and checked: X'C0' and X'00'.

Note: Before the routine selection, all the signals must be wrapped together according to NTT specifications.

ERC	RAC	Error Description
0611	*A0	LIC Modem-In register does not contain the expected value (X'50' to X'00').

QD05 - LIC LINE TEST - MANUAL INTERVENTION ROUTINE FOR JAPAN

(LIC 4 NTT Modem-In Wrap Test)

This routine is used to perform a wrap test on the modem control leads using the NTT wrap block located at the external cable end for a LIC 4 line. The test takes place at installation time at the request of the NTT maintenance personnel.

FUNCTION: Write a pattern into the LIC Modem-Out register of the line and read the contents of the wrapped pattern into the LIC Modem-In register of the same line.

Then compare the Modem-In register content to the expected pattern. 2 patterns are sent and checked: X'C4' and X'00'.

Note: Before the routine selection, all the signals must be wrapped together according to NTT specifications.

ERC	RAC	Error Description
0611	*A0	LIC Modem-In register does not contain the expected value (X'40' to X'00').

RA01 - ICC CARD TEST: ICC ADDRESS BUS PARITY CHECKER TEST

This routine checks that the parity checker of the address bus in each ICC card is error free. The ICC type is also verified

FUNCTION: All ICC registers (00-01-10) for the 16 lines connected to the ICC are addressed in read mode with good and bad parity.

STEP:

1. If parity is OK: Expected result: read operation complete (result given in FES XR16).
2. If parity not OK: Expected result: read operation not complete with ICC error (result given in FES XR16).
3. The ICC type read from the ICC card is compared with the value given in the CDF.

ERC	RAC	Error Description
0611	*90	Read operation not complete (with parity OK, Step 1)
0621	*90	Read operation complete (with parity not OK, Step 2)
0630	*97	Mismatch between the ICC type in ICC card and in CDF (Step 3) (with parity OK).

RA02 - ICC CARD TEST: 4C RAM PARITY CHECKER TEST

This routine ensures that the parity checker following the 4C RAM is working correctly in each ICC card.

FUNCTION: Four patterns are sent on the data bus to the 4C RAM of the first line (not autocall and for a LIC 1 line) connected to the ICC under test.

Using the patterns: X'08', X'F8', and X'58' (with good parity), and X'A0' (with bad parity). Each entry of the parity checker can be activated with X'1' and X'0'.

STEP:

1. 4C RAM parity checker in error (with good parity).
2. 4C RAM parity checker in error (with bad parity).

ERC	RAC	Error Description
0611	*90	Interrupt level 2 with ICC check condition occurs erroneously (good parity, Step 1).
0621	*90	Interrupt level 2 with ICC check condition does not occur (bad parity, Step 2).

RA03 - ICC CARD TEST: 4D RAM (BCCW) PARITY GENERATOR TEST

This routine checks that the parity generator of the BCCW (bit clock counter word) in each ICC card is error free.

FUNCTION: In order to activate the BCCW parity generator, four patterns (2 with good parity and 2 with bad parity) are sent on the data bus from the FES card to the 4D RAM in each ICC card. Each pattern is sent to the 4D RAM of the first line (not autocall and for a LIC 1 line) connected to the ICC under test.

STEP:

1. 4D RAM parity checker in error (with good parity).

2. 4D RAM parity checker in error (with bad parity).

ERC	RAC	Error Description
0611	*90	Interrupt level 2 with ICC check condition occurs erroneously (good parity, Step 1).
0621	*90	Interrupt level 2 with ICC check condition does not occur (bad parity, Step 2).

RA04 - ICC CARD TEST: ICC RAMS GATING TEST

This routine checks that ICC RAM selection mechanism is working correctly in each ICC card.

FUNCTION: Write various patterns in each ICC RAM entry of the first line connected to the ICC card. The entries are then read back and compared with the written patterns. The patterns used are: 4D1: X'94', 4D2: X'28', 4C : X'67'.

Write the above 3 patterns again, but in the reverse sequence, from 4C RAM to 4D1 RAM and do the same checking.

The aim of this routine is to check the correct gating between the 4D1, 4D2, and 4C RAMs.

ERC	RAC	Error Description
0611	*95	4D1, 4D2, or 4C ICC RAMs do not contain the expected value for the test done.
Note.This error can occur on first detection of CLCK 4 signal grounded		

RA05 - ICC CARD TEST: ICC RAMS ADDRESSING TEST

This routine checks that each ICC RAM entry is correctly selected in each ICC card.

FUNCTION: Write in each ICC RAM entry a pattern with its relevant address as data. First run from line address 0 to line address 15 and check and report errors, if any. Repeat for line address 15 to line address 0, but with the inverted address as data, and check and report error if any. This process is done RAM-by-RAM (4D1-4D2-4C).

ERC	RAC	Error Description
0611	*90	One ICC RAM entry does not contain the expected value for the test done.

RA06 - ICC CARD TEST: ICC RAMS VALIDITY TEST

This routine checks that each bit of the ICC RAMs can be set with 0s and 1s.

FUNCTION: ICC RAM entries are written with a specific pattern: X'AA', then X'55'.

Then each entry is read back and compared with the sent pattern. This process is done RAM-by-RAM (4D1-4D2-4C).

ERC	RAC	Error Description
0611	*90	One ICC RAM entry does not contain the expected value for the test done.

RA07 - ICC CARD TEST: ICC RAM RESET TEST

This routine ensures that the 'Reset Latches' lead is correctly handled by the 4C RAM in ICC the card.

FUNCTION: The 4C RAM, on the ICC card, written with ones. The 'reset latches' command is activated in the FES card and the ICC 4C RAM contents are checked to be X'0'.

ERC	RAC	Error Description
0611	*90	One (or more) entries of ICC 4C RAM is not reset.

RB01 - ICC LINE TEST: ERROR ADDRESS REGISTER TEST

This routine checks that the error address register is working correctly for all the LIC 1 lines (other than autocall) connected to the ICC card.

FUNCTION: Write a pattern with bad parity on the line entry under test of the 4C RAM on the ICC card.

The result in the TSS interrupt handler level 2 is checked for an 'ICC check'.

ERC	RAC	Error Description
0611	x90	Interrupt level 2 with ICC check condition does not occur for corresponding line.

RB02 - ICC LINE TEST: BIT CLOCK COUNTER WORD INCREMENT FUNCTION TEST

This routine ensures that the increment function of the BCCW is working correctly for each line of the 4D RAM in the ICC card.

FUNCTION: Seven patterns are sent on the data bus from the FES card to each entry under test of the ICC 4D RAM in order to activate the increment function of the BCCW. The entry under test is then read back and the BCCW read patterns compared with the BCCW expected patterns.

STEP:

1. Error in ICC 4D1 RAM increment

2. Error in ICC 4D2 RAM increment

ERC	RAC	Error Description
0611	x90	4D1 does not contain the expected value for the test done (Step 1).
0612	x90	4D2 does not contain the expected value for the test done (Step 2).

RB03 - ICC LINE TEST: BCCW CORRECTION ALGORITHM TEST IN SYNCHRONOUS MODE

This routine ensures that the correction algorithm, performed to adjust the receive clock mechanism, is working correctly in the synchronous mode for each line entry of the 4D RAM on the ICC card.

FUNCTION: In the synchronous mode, 5 types of correction on the BCCW value are provided by the ICC card according to the BCCW value when a 'line state change' occurs: (+6, +4, -2, -4, and no change). Nine patterns are sent on the data bus, from the FES card to the line under test entry of the 4D RAM in order to check the correction algorithm mechanism in the synchronous mode.

Two patterns are sent for each correction possibility, depending on the BCCW counter value. The correction is activated by 'line change latch'. The checking is done by comparing the read and expected patterns in the BCCW.

STEP:

1. Error on 4D1 RAM during correction algorithm activation (in synchronous mode).

2. Error on 4D2 RAM during correction algorithm activation (in synchronous mode).

ERC	RAC	Error Description
0611	*90	4D1 does not contain the expected value for the test done (Step 1).
0612	*90	4D2 does not contain the expected value for the test done (Step 2).

RB04 - ICC LINE TEST: BCCW CORRECTION ALGORITHM TEST IN ASYNCHRONOUS MODE

This routine ensures that the correction algorithm, performed to adjust the receive clock mechanism, is working correctly in the start/stop mode, for each line entry of the 4D RAM on the ICC card.

FUNCTION: In start/stop mode, when a line state change occurs, the corresponding BCCW is always forced to X'33'. Three patterns (all 0s, all 1s, no change) are sent on the data bus from the FES card to the line under test entry of the 4D RAM in order to check the correction algorithm mechanism in S/S. The BCCW counter value of each read pattern is expected to be set to X'33'.

STEP:

1. Error on 4D1 RAM during correction algorithm activation (in start/stop mode).

2. Error on 4D2 RAM during correction algorithm activation (in start/stop mode).

ERC	RAC	Error Description
0611	*90	4D1 does not contain the expected value (Step 1).
0612	*90	4D2 does not contain the expected value (Step 2).

RC01 - ICC/LIC TEST: INTERNAL CLOCK TEST

This routine ensures that the receive and transmit clocks provided by the ICC card, in 'Internal Clocking Mode', are correctly working on the ICC and LIC 1 cards. Only the LIC 1 line is exercised by the routine (except autocall).

FUNCTION: A 2-byte pattern initiated in CSP card is sent to the LIC 1 line (not autocall). The line is defined in 'internal clocking mode'. In the relevant line, the transmit data is wrapped with receive data (using LIC internal wrap facility).

The 4C RAM entry on ICC card is set to send a clock at 50 bps in start/stop mode and the FES card is activated.

Comparison is done between CSP data burst transmit and receive. The results are analyzed on an ICC card basis.

ERC	RAC	Error Description
		CSP data burst receive located in CSP RAM address X'8010' is not equal to CSP data burst transmit in CSP RAM address X'8000'.
0611	*87	Error between ICC and one LIC 1 having at least one line not autocall.
0611	*92	Error between ICC and several LIC 1s having at least one line not autocall.
Note: ADDIT INFO, displayed on the screen, flags the lines (depending on ICC card) found in error.		

RC02 - ICC/LIC TEST: LOCAL ATTACHMENT CLOCK TEST

This routine checks that the local attachment clock provided by the ICC card is correctly working on the ICC and related LIC cards. All lines (except autocall) are exercised by the routine.

FUNCTION: A 2-byte pattern initiated in CSP card is sent to the LIC line (not autocall). The line is defined in 'local attachment mode'. In the relevant line, the transmit data is wrapped with receive data (using LIC internal wrap). The FES card is activated for a transfer to a slower speed (2400 bps). Comparison is done between CSP data burst transmit and receive. The results are analyzed on an ICC card basis.

ERC	RAC	Error Description
		CSP data burst receive located in CSP RAM address X'8010' is not equal to CSP data burst transmit in CSP RAM address X'8000'.
0611	*88	Error between ICC and one LIC 1 having at least one line not autocall.
0611	*94	Error between ICC and several LIC 1s having at least one line not autocall.
Note: ADDIT INFO, displayed on the screen, flags the lines (depending on ICC card) found in error.		

RC03 - ICC/LIC TEST: INTERNAL CLOCK DIVIDER EXERCISING TEST

This routine ensures that the internal clock dividers on the ICC card are providing a clock. Only the first LIC 1 line (except autocall) is exercised by the routine.

FUNCTION: A 2-byte pattern initiated in CSP card is sent to the first non-autocall line connected to a LIC 1.

The corresponding line is defined in 'internal clocking mode' and transmit data is wrapped with receive data (using the LIC internal wrap facility).

In the ICC card the 4C RAM entry of the line under test is successively set with 1200, 134.5, 110, and 50 bps in start/stop mode.

The FES is activated for the corresponding time. Comparison is done between CSP data burst transmit and receive.

ERC	RAC	Error Description
		CSP data burst receive located in CSP RAM address X'8010' is not equal to CSP data burst transmit in CSP RAM address X'8000'.
0611	*90	Internal clock 1200 bps divider not working.
0612	*90	Internal clock 134.5 bps divider not working.
0613	*90	Internal clock 110 bps divider not working.
0614	*90	Internal clock 50 bps divider not working.

RC04 - START/STOP: OLD CORRECTION MECHANISM

This routine verifies that the Start/stop is working for the following frequencies: 75, 100, 200, 300, and 600 bps.

FUNCTION: A 2-byte pattern initiated in CSP card is sent to the first non-autocall line connected to a LIC 1.

The corresponding line is defined in 'internal clocking mode' and transmit data is wrapped with receive data (using the LIC internal wrap facility).

In the ICC card the 4C RAM entry of the line under test is successively set with 75, 100, 200, 300, and 600 bps in Start/Stop mode.

The FES is activated and data is expected in the delay corresponding to the selected speed. Comparison is done between the transmit and receive burst.

ERC	RAC	Error Description
		Received CSP data burst is not equal to transmitted CSP data burst.
0611	*90	75 bps is not working
0612	*90	100 bps is not working
0613	*90	200 bps is not working
0614	*90	300 bps is not working
0615	*90	600 bps is not working

RC05 - START/STOP: NEW CORRECTION MECHANISM

This routine verifies that the Start/stop is working for the following frequencies: 2400, 4800, 9600 and 19200 bps.

FUNCTION: A 2-byte pattern initiated in CSP card is sent to the first non-autocall line connected to a LIC 1.

The corresponding line is defined in 'internal clocking mode' and transmit data is wrapped with receive data (using the LIC internal wrap facility).

In the ICC card the 4C RAM entry of the line under test is successively set with 2400, 4800, 9600, and 19200 bps in Start/Stop mode.

The FES is activated and data is expected in the delay corresponding to the selected speed. Comparison is done between the transmit and receive burst.

BCCW is set to 48 at 2400 bps, and to 56 at 4800, 9600, and 19200 bps.

ERC	RAC	Error Description
		Received CSP data burst is not equal to transmitted CSP data burst.
0611	*90	2400 bps is not working
0612	*90	4800 bps is not working
0613	*90	9600 bps is not working
0614	*90	19200 bps is not working

XXXX - TSS DIAGNOSTICS - LEVEL 1 INTERRUPT HANDLER REPORTING

Level 1 is used by the TSS diagnostics as a software interrupt defined to report:

- Errors occurring during a TSS asynchronous access (result given in FES XR16).
- Errors occurring during a TSS synchronous general command (result given in FES XR17).

The following errors can occur when an unexpected level 1 occurs in the CSP during a TSS diagnostic routine (Pxxx - Qxxx - Rxxx).

ERC	RAC	Error Description
D0X1	*50	FES internal error. Asynchronous access XR16 bits 1 and 5 on.
D0X2	*60	LIC error. Asynchronous access. XR16 bits 1 and 4 on. (Note: First detection of FES-LIC-ICC address bus grounded)
D0X3	*60	ICC error. Asynchronous access. (XR15 is addressing a ICC register) (Note: First detection of FES-LIC-ICC address bus grounded)
D0X4	*67	Synchronous error during a general command XR17 bit 4 on (during reset command). (Note: First detection of clock 1-2-3 signals grounded in TSS).

XXXX - TSS DIAGNOSTICS - LEVEL 2 INTERRUPT HANDLER REPORTING

The following errors can occur when an unexpected level 2 occurs in the CSP during a TSS diagnostic routine (Pxxx - Qxxx - Rxxx).

ERC	RAC	Error Description

		Following information is displayed on the screen:
		- ERRBIT field
		  - First byte = xx
		  FES XR12 value (line interface address for interrupt level 2)
		- Second byte = xx
		  FES XR10 value (extended IRR register)
		- IAR of PSW level 7
		- A message giving (in hexadecimal) the line interface number
		  (transmit or receive) having caused the unexpected interrupt.

E0X1	*50	Normal data process interrupt.
E0X2	*5B	Underrun or overrun condition.
E0X3	*50	Timeout condition.
E0X4	*63	Modem Change condition.
E0X5	*50	FES/LIC error (in synchronous mode).
E0X6	*50	FES internal error.
E0X7	*63	LIC driver check condition.
E0X8	*62	ICC check (in synchronous mode).
E0X9	*53	CSP/FES error.
E0XA	*5B	Underrun front end sequence error Underrun is detected by front end layer with TE bit on.
E0XB	*5B	End of Transmission (EOT) condition.
E0XC	*66	Ending flags condition (see note) (used to report line protocol state or error).
Note: First detection of several errors found with ICC/LIC tests.		

XXXX - TSS DIAGNOSTICS - LEVEL 0 INTERRUPT HANDLER REPORTING

The following errors can occur when an unexpected level 0 occurs in the CSP during a TSS diagnostic routine (Pxxx - Qxxx - Rxxx).

ERC	RAC	Error Description

		Following information is displayed on the screen:
		- ERRBIT field
		  - First byte = xx
		  CSP XR03 value (error register)
		  - Second byte = bbbb 0000
		  bbbb = error condition saved in PSW of interrupted level.
		- PSW level 7 = IAR of PSW level 7.
		- LI = IAR of interrupted level.

F001	*57	Unexpected adapter acknowledge.
F002	*26	Control store data check in write operation.
F003	*28	CSP check with control store data check condition.
F004	*52	CSP check with LSR-XR parity check condition.
F005	*20	CSP check with CSP internal check condition.
F006	*25	External register address check.
F007	*27	Control store address check.
F008	*20	Local store address check.
F009	*57	Adapter Interconnection check.
F00A	*57	External adapter check.
F00B	*20	CSP check with type not indicated by hardware.
F00C	*53	Multiple CSP check (too many bits in PSW bit configuration).

CHAPTER 6. MOSS DIAGNOSTICS

Table of Contents for Chapter 6

CHAPTER 6. MOSS DIAGNOSTICS

This Chapter describes the tests carried out to check the MOSS during IML.

IML CHECKOUT - ROS PART

This test is entirely in ROS.

MPC CARD TEST (NO DISPLAY)

During this series of tests, the hexadecimal indicators remain blank; only the MPC card is tested.

Basic MPC Instruction Test Part 1

This routine checks the following instructions:

- LRI — Load Register Immediate
- TRI — Test Register Immediate
- BALR — Branch and Link Register
- SR — Subtract Register
- LHR — Load Halfword Register
- SRI — Subtract Register Immediate
- CR — Compare Register
- AR — Add Register
- AYR — Add Register with Carry
- SYR — Subtract Register with Carry
- BZR — Branch on Register Equal 0
- SLL — Shift Left Byte Logical
- SLHL — Shift Left Halfword Logical
- BCTR — Branch on Count Register
- BCR — Branch on Condition Register
- ARI — Add Byte Register Immediate
- KI READ MCPC
- KI WRITE MCPC
- KI WRITE PP
- KI READ PP

and the following extended jump mnemonic operations:

- After an arithmetic instruction:
  - Jump if Minus
  - Jump if Not Plus
  - Jump if Not Minus
  - Jump if Carry
  - Jump if Zero
  - Jump if Not Zero
- After compare instructions:
  - Jump if Not Equal
  - Jump if High
  - Jump if Not Low
- After a test instruction:
  - Jump if Equal to Mask
  - Jump if Not Mixed Under Mask
- After any instruction:
  - Jump Unconditionally

Important Note: if an error occurs, the program hangs without any display.

ROS CRC 16 Calculation, Instruction Test Part 2

This routine checks the following instructions:

- ORI — OR Byte Register Immediate
- XRI — Exclusive OR Byte Register Immediate
- LHR — Load Halfword Register
- SHR — Subtract Halfword Register
- AHR — Add Halfword Register
- NR — AND Byte Register

- OR OR Byte Register
- RL Rotate Left Byte Register
- ARI Add Byte Register Immediate
- XR Exclusive OR Byte Register
- LR Load Byte Register
- RLH Rotate Left Halfword Register
- NRI AND Byte Register Immediate
- After a test instruction:
  - Jump if not all zero
- After an arithmetic instruction:
  - Jump if plus
- After a compare instruction:
  - Jump if equal
  - Jump if high
  - Jump if low
  - Jump if not equal
  - Jump if not high
  - Jump if not low
- After a logical instruction:
  - Jump if all zero
  - Jump if all ones
  - Jump if mixed
  - Jump if not all zero
  - Jump of not all ones
  - Jump if not mixed

The CRC 16 calculation is made between the address X'0000' and the label: CHGROCRC. This label contains a pre-calculated ROS CRC 16.

Important Note: if an error occurs, the program hangs without any display.

Register Space Test

This routine checks the register space in two passes:

- Pass 1 tests the parity checker.
- Pass 1 tests the registers in the range X'000' through X'2FF'.
- Pass 2 tests the registers in the range X'300' through X'3FF'.

Operation:

The pass 1 tests run under the control of the ROS test controller.

The pass 2 tests run in stand alone mode.

Important Note: if an error occurs, the program hangs without any display.

Internal PIO Bus Test, PKD Adapter Test, in Diagnostic Mode

This test checks PKD module accessibility via the internal PIO bus. It also checks the correct decoding of the PKD command set, and that the PKD registers are reset.

Important Note: if an error occurs, the program hangs without any display.

Internal PIO-BUS/PKD Adapter Test in Wrap Mode

This routine places floating 1s and 0s on the data bus to complete PIO bus checking.

Tests patterns (floating 1 and floating 0) are written into the PKD module previously set up in the wrap mode, followed by a read and compare for each test pattern. At the end of the routine, an MCPC checking is done.

Important Note: if an error occurs, the program hangs without any display.

Hexadecimal Display Test in Wrap Mode

Data is written in the wrap mode into the hexadecimal display registers to check if any data can be displayed:

1. A test pattern is written into hexadecimal display register (AB) and saved.
2. The test pattern is written again into the same hexadecimal register to push the first one into the serial sense input register.
3. A read is done of the sense register to compare it with the saved test pattern.
4. A new test pattern is created and the sequence is restarted.
5. When complete (test data = X'00') the same test is done on hexadecimal display register C

If an error occurs, the program loops on the faulty hexadecimal display value.

IML STEP F00

1. Send 'F00' to the display.
2. Set the message lamp on
3. Read the message bit, and set an error E60 if not on.
4. Read the rotary image, and set an error E60 if the received data = 00.
5. If the IML type is presumed to be POR, and the MOSS IMLed switches are off, then the POR bit must be present in the rotary image. If the bit is not on, then set an error E63.
6. Reset the power on remember bit and the start bit.
7. Check for a correct reset of these bits; if they are not reset, then set an error E63.
8. Check that there is one, and only one, bit in the IML type requested; if not, set an error E64.
9. If the POR and the start bits are both on in the rotary image, then set an error E63.
10. If the POR and the start bits are both off, check if it is a software request; if it is not, set an error E66. If it is a software request, restore the previous IML type in place of the read type, then go to step 13.
11. If start was requested, the start bit flipped during the memory test, and the MOSS IMLed switch and CCU running status were both off, then replace the start bit by the POR bit in the requested IML type.
12. If a console link test was requested, set the IML diagnostic type.
13. Read the channel adapter interface B status register. If the received data = 00, then set an error E60.
14. Read the channel adapter interface A status register. If the received data = 00, then set an error E60.
15. Test if the CE jumper is present, if yes, then set an error E61.
16. Test if the loop on error jumper is present, if yes, then set an error E61.
17. If the IML type is POR and a dump was requested, then set an error E65.
18. If a POR IML was requested, check the channel adapter statuses; they must be all disabled at POR. If not, then set an error EE1 through EEA depending on the CA address and the faulty interface.
19. End step F00 and go to step F01.

The following table resumes the different error codes possible during IML step F00:

Hex Display	Error Description
E60	- not possible to set message lamp - rotary image = 00 - CA interface A status register = 00 - CA interface B status register = 00
E61	- CE jumper present at IML time - loop on error jumper present at IML time
E63	- discrepancy between memory status and POR bit in in rotary image - not possible to reset POR or start bit in rotary image register - POR and start bits are both present in rotary image
E64	- No bit or more than one bit in rotary image IML type part
E65	- POR IML decoded and MOSS dump requested, operator error
E66	- POR and start bits are both off in rotary image, and no software request in software switches
EE1	CA1 interface A enable at power on
EE2	CA2 interface A enable at power on
EE3	CA3 interface A enable at power on
EE4	CA4 interface A enable at power on
EE5	CA5 interface A enable at power on
EE6	CA6 interface A enable at power on
EE7	CA1 interface B enable at power on
EE8	CA2 interface B enable at power on
EE9	CA3 interface B enable at power on
EEA	CA4 interface B enable at power on

IML STEP F01 (POWER ON CASE)

1. Set F01 on hexadecimal display.

2. Test register space part 1 (X'0000' through X'0300'); all register locations may be erased at POR.

3. Write X'FF' into all locations of register space part 1.

4. Read a register space location, and check that the received data = X'FF', if not, set an error E80.

   Set X'80' into the same location. Loop on step 3 until all locations are tested.

5. Read a register space location, and check that the received data = X'80', if not, set an error E80, then set X'00' into the same location and loop on step 4 until all locations are tested.

6. Read a register space location, and check that the received data = X'00', if not, set an error E80. Then again set X'00' into the same location, and loop on step 5 until all locations are tested.

7. Basic storage test for addresses X'2000' through X'FFFF'.

8. Write X'FF' into all basic storage space.

9. Read a storage location, and check that the received data = X'FF', if not go to step 12 to decode the error number. Then set X'80' into the same location, and loop on step 8 until all the storage space locations have been tested.

10. Read a storage location, and check that the received data = X'80', if not go to step 12 to decode the error number. Then set X'00' into the same location, and loop on step 9 until all the storage space locations have been tested.

11. Read a storage location, and check that the received data = X'00', if not go to step 12 to decode the error number. Then set X'00' into the same location again, and loop on step 10 until all the storage space locations have been tested.

12. Read 1 storage location, check if received data = 00 if no go to step 12 to decode error number set again 00 in same location, loop on step 10 until all storage space locations are tested

13. Check if the error occurred in the pluggable RAM module (X'A000' through X'FFFF').

    - if no, set an error E81.
    - if yes, set an error E82

14. End step F01 (power on case) and go to step F03.

The following table resumes the different error codes possible during IML step F01 (power on case):

Hex Display	Error Description
E80	During a Power on IML, an error was found during the register space part 1 test (X'0000' through X'0300').
E81	An error was found during the basic storage test on the fixed module located at X'2000' through X'BFFF'.
E82	An error was found during the basic storage test on the pluggable module located at X'A000' through X'FFFF'.

IML STEP F01 (NON-POWER ON IML)

1. The current PSW is in the register space at locations X'0000' through 001F; save it at the end of the register space (X'03E0' through X'03FF').
2. Perform the read only test in basic storage (X'2000' through X'2500' (operational program communication area).
3. Perform the basic storage test (X'2500' through X'FFFF') then go to step 8.

IML STEP F03

In this test, if an error is encountered, the test is not stopped, but an error bit is set in an error switch; the run continues on the next test. When all tests have been run, an error code is returned in the range EC1 through ECF. The last digit is an error code, whose bit meaning is:

Bit	Meaning
8	Error in MMC/MPC interconnection
4	Error in MCC/MPC interconnection
2	Error in CCA/MPC interconnection
1	Error in DAC/MPC interconnection

1. Set the hexadecimal display to F03.
2. Set the hexadecimal display to EC0 (not blinking).
3. Read the last ROS address contents and save the value.
4. Enable the external PIO bus; if an error is found either on the data bus or on the address bus, the program loops, and the display EC0 is seen indefinitely.
5. Prepare the TTA level 0 block '0'. After enabling the MEF (storage expansion feature), this allows level 0 block '0' to generate a virtual address of X'00000'. Check correct TTA writing; set the storage interconnection error bit if it is not possible to write in the TTA.
6. Prepare TTA level 0 block 'F' with a value of '01'. After enabling the MEF, this allows level 0 block 'F' to generate a virtual address of X'01FFF'. This is the last ROS address, the contents of which were saved in 3 above; check the TTA value, if not correct, set the storage interconnect error bit.
7. Enable the storage expansion feature.
8. Using the MEF, read the last ROS address contents, and compare against the known value; if not equal, set the storage interconnection error bit.
9. Using the MEF, read the contents of the first ROS address; it must be X'FFFF', if not, set storage interconnect error bit.
10. If any MCPC bit is on, or the IOIRR bit 0 is on, set the storage interconnect error bit.
11. Disable the MEF.
12. Sense the MCC status register. Set the MCC interconnect error bit if any MCPC bit on, the reset the MCPC.
13. Sense the CCA status register. Set the CCA interconnect error bit if any MCPC bit on, then reset the MCPC.
14. Sense the DAC status register. Set the DAC interconnect error bit if any MCPC bit on, then reset the MCPC.
15. If error switch 0 is on, return an error code EC1/ECF.
16. End step F03 and go to step F04.

Note: when an error code is returned by this test (except for EC0 which is a catastrophic error), it is possible to loop on the error and scope the flow of this test. See the MIM part 3 for the complete loop on error description.

IML STEP F04

1. Set F04 on the hexadecimal display.

2. Disk adapter test phase 1.

   a. Reset the disk adapter.
   b. Read the sense byte to check the parity valid tag; if any MCPC bit is on, set an error E70.
   c. Set the machine check bit in basic status.
   d. Read the basic status to check the machine check bit; if not on, set an error E70.
   e. Reset the machine check bit.

3. Disk adapter test phase 2.

   a. Set the enable interrupt, operation complete, and head engaged bits in the status register.
   b. Read the status register and check that the expected bits are present; if not, set an error E71.
   c. Check if a level 5 interrupt request is present in the IOIRR; if not set an error E71.
   d. Reset the operation complete and head engaged bits in the status register.
   e. Read the status register; only the enable interrupt bit must be on, if any error set a code E71.
   f. No level 5 interrupt request must be present in the IOIRR, if any error, set a code E71.

4. Disk adapter test phase 3.

   a. Write address X'AAAA' in the MAR (storage address register).
   b. Read the MAR, if incorrect result set a code E72.
   c. Write address X'5555' in the MAR.
   d. Read the MAR, if incorrect result set a code E72.
   e. Clear the MAR.

5. Disk adapter test phase 4.

   a. Set the enable interrupt and enable timer bits in the status register.
   b. Start a 4ms software timer, and wait for timeout.
   c. Read the status register, and check that there is no timer interrupt. If a timer interrupt is present, set an error E73.
   d. Start a 2ms software timer, and wait for timeout.
   e. Read the status register. A timer interrupt must be present, if not set an error E73.
   f. Reset all the status register bits, except for the enable interrupt bit.

6. Disk adapter test phase 5.

   a. Enable the index mark interrupt bit in the status register.

   b. Reset the index mark interrupt, if any.

   c. Start a 2-second software timer and wait for a level 5 interrupt request.

   d. When the level 5 interrupt occurs, check the index mark bit. If it is not present, set an error E74.

   e. If there is no level 5 interrupt before the timer times out, set an error code E74.

   f. Disable the index mark bit and reset the interrupt.

   Note: a full diskette speed test is made by the disk CAC during step F05.

7. Disk adapter test phase 6.

   a. Set the MFM bit in the sector identifier register.

b. Read the sense byte and check that the MFM bit is on, if not, set an error code E75.

8. IOIRR Test. Check that no interrupt is pending. If an interrupt is present, set an error code EAx for levels 0 through 3, or EBx for levels 4 through 7.

9. Interrupt mechanism test.

   a. Starting at level 0, set the PIRR = X'FF'.
   b. Read the PIRR. If not = X'FF' set an error code E80.
   c. Set the level to 1.
   d. Mask all levels except the current level in the common mask.
   e. Check the actual level against the expected level. If there is an error set a code E80.
   f. Write the common mask to mask all levels except level 0, and check for correct level 0 return.
   g. Increment the level, and return to step 'd' to exercise all levels from 2 to 7. When all levels have been exercised, leave the interrupt mechanism test.

10. Set step F05 and give control to the disk support adapter program, which loads the RAM part of the MOSS diagnostics, and then returns to IML Step F06 below.

The following table resumes the different error codes possible during IML step F04:

Hex Display	Error Description
E70	- MCPC error when DAC sense byte is read - unable to exercise machine check bit in DAC basic status register
E71	- error in set or reset of DAC status register - no level 5 request when interrupt request bit is set in DAC status register. - level 5 request always present after reset of interrupt request bit in DAC status register.
E72	- read pattern in DAC storage address register does not match expected pattern
E73	- DAC 5ms timer is out of 20% tolerance
E74	- diskette not running or not present
E75	- unable to set MFM mode
E80	- interrupt mechanism does not work
EA1	- permanent interrupt request on level 3
EA2	- " " " " " 2
EA3	- " " " " " 2 and 3
EA4	- " " " " " 1
EA5	- " " " " " 1 and 3
EA6	- " " " " " 1 and 2
EA7	- " " " " " 1, 2, and 3
EA8	- " " " " " 0
EA9	- " " " " " 0 and 3
EAA	- " " " " " 0 and 2
EAB	- " " " " " 0, 2, and 3
EAC	- " " " " " 0 and 1
EAD	- " " " " " 0, 1, and 3
EAE	- " " " " " 0, 1, and 2
EAF	- " " " " " 0, 1, 2, and 3
EB1	- permanent interrupt request on level 7
EB2	- " " " " " 6
EB3	- " " " " " 6 and 7
EB4	- " " " " " 5
EB5	- " " " " " 5 and 7
EB6	- " " " " " 5 and 6
EB7	- " " " " " 5, 6, and 7
EB8	- " " " " " 4
EB9	- " " " " " 4 and 7
EBA	- " " " " " 4 and 6
EBB	- " " " " " 4, 6, and 7
EBC	- " " " " " 4 and 5
EBD	- " " " " " 4, 5, and 7
EBE	- " " " " " 4, 5, and 6

IML CHECKOUT, RAM PART

IML STEP F06

BASIC CCA TEST

1. Valid command recognition.

   a. Point to valid command table.
   b. Send a valid command.
   c. If an error occurs, display E50.
   d. Loop on the next command until end of table.

2. Invalid commands test.

   a. Send all possible command values from X'00' to X'FF' which are not in the valid command table.
   b. Check after each command machine check bit is present in the basic status; if not present, display E50.
   c. Reset the machine check bit in the basic status register and check that it is correctly reset; if not reset, display E50.
   d. Loop on the invalid command test until all possible values have been sent.

3. Adapter control register test.

   a. Check that the adapter control register is set to all '0's at routine entry; if not, display E50.
   b. Set the adapter control register using X'55' and check that the result is X'55'; if not, display E50.
   c. Set the adapter control register using X'AA' and check that the result is X'FF' (X'55' + X'AA' = X'FF'); if not, display E50.
   d. Reset the adapter control register using X'55' and check that the result is X'AA' (X'FF' - X'55' = X'AA'); if not, display E50.
   e. Reset the adapter control register using X'AA' and check that the result is X'AA' (X'AA' - X'AA' = X'00'); if not, display E50.

4. Modem control register test.

   a. Read the modem control register. It must contain X'00' at routine entry; if not, display E50.
   b. Write the modem control register with X'55'.
   c. Read the modem control register. It must be equal to X'55'; if not equal to X'55', display E50.
   d. Write the modem control register with X'AA'.
   e. Read the modem control register. It must be equal to X'AA'; if not equal to X'AA', display E50.
   f. Write the modem control register with X'00'.
   g. Read the modem control register. It must be equal to X'00'; if not equal to X'00', display E50.

5. Modem status register test.

   a. Set the WRAP mode in the modem control register.
   b. Reset any pending modem interrupts in the basic status.
   c. Set the modem control register to get a DSR transition.
   d. Read the basic status to check the modem interrupt and interrupt request; if not display E50.
   e. Read the modem status register to check the DSR transition, if not, display E50.
   f. Reset the modem interrupt in the basic status.
   g. Send the basic status to check for correct reset; if not reset, display E50.
   h. Set the modem control register to get a CTS transition.
   i. Read the basic status to check the modem interrupt and interrupt request; if not display E50.

j. Read the modem status register to check the CTS transition; if not, display E50.
k. Reset the modem interrupt in the basic status.
l. Read the basic status to check for a correct reset, if not display E50.
m. Set WRAP and DTR in the modem control register.
n. Read the basic status to check the modem interrupt and interrupt request; if not, display E50.
o. Read the modem status register to check the DSR and CTS transitions; if not, display E50.
p. Reset the modem interrupt in the basic status.
q. Reset the DTR in the modem control register.
r. Read the basic status to check the modem interrupt; if not, display E50.
s. Read the modem status register to check the DSR transition; if not, display E50.
t. Reset the adapter.
u. Set the WRAP mode in the modem control register.
v. Set the transmit mode in the adapter control register.
w. Read the modem status register to check the CTS transition; if not, display E50.

6. Timer test.

a. Load the timer register for a 1.6ms delay.
b. Software wait of 1.4ms.
c. No timer interrupt must be present in the basic status; if present, display E50.
d. Software wait of 0.4ms.
e. A timer interrupt must be present in the basic status; if not present, display E50.
f. Reset the timer interrupt in the basic status register.
g. Check for correct reset in the basic status; if not reset, display E50.
h. Load the timer register for a 0.2-second timeout.
i. Software wait of 180ms.
j. No timer interrupt must be present in the basic status; if present, display E50.
k. Software wait of 40ms.
l. A timer interrupt must be present in the basic status; if not present, display E50.
m. Set the timer high register with X'7F' (bit 0 off must cause a timer interrupt in the basic status).
n. Read the basic status register to check that a timer interrupt is not present; if present, display E50.
o. Set the timer high register with X'FF' (means start a 25 seconds timer).
p. Set the timer low register with X'01' (means reset the timer high register).
q. A timer interrupt must occur before 1ms; if not, display E50.
r. Reset the timer interrupt in the basic status register.
s. Set the timer low register with X'01' (means start a 0.78ms timer).
t. Set the timer high register with X'00' (means stop the timer).
u. No timer interrupt must occur during a 1ms software wait; if a timer interrupt occurs, display E50.
v. Reset the timer interrupt in the basic status.
w. Set the timer high register with X'80' (means immediate time out).
x. A timer interrupt must be present in the basic status; if not present, display E50.

7. Output request and receive clock run.

a. Set the WRAP mode in the modem control register.
b. Load the transmit buffer with the character to send.
c. Set the receive and transmit mode in the adapter control register.
d. An output request interrupt must happen before a software time out; if there is no interrupt, display E50.
e. Load the transmit buffer with a second character.
f. The output request interrupt must be off in the basic status register; if not, display E50. Read the adapter

status register to check that the receive clock is running; if not present, display E50.

8. Input request.

   a. Set the WRAP mode in the modem control register.
   b. Load the transmit buffer with the character to send (X'08').
   c. Set the receive and transmit mode in the adapter control register.
   d. Software loop to wait for interrupt.
   e. At timeout, check that output and input requests are present in the basic status register; if not present, display E50.
   f. Read the received character, and check that the correct data was received; if not, display E50.
   g. Read the basic status to check reset of input request; if not reset, display E50.

9. Test that the input request is blocked if the receive mode is off.

   a. Set the WRAP mode in the modem control register.
   b. Load the transmit buffer with the character to send.
   c. Set the transmit mode in the adapter control register.
   d. Software loop to wait for a possible interrupt.
   e. At timeout check that there is no input request in the basic status register; if present, display E50.

10. Overrun bit test.

   a. Set the WRAP mode in the modem control register.
   b. Load the transmit buffer with the character to send.
   c. Set the receive and transmit mode in the adapter control register.
   d. Software loop to wait for interrupt request.
   e. At timeout, check for output and input request in the basic status register; if not present, display E50.
   f. Load the transmit buffer to send another character.
   g. Software loop to wait for end of transmission.
   h. Read the basic status to check for output request, input request, exception request and interrupt request; if not all present, display E50.
   i. Read the adapter status register to check for the overrun bit; if not present, display E50.
   j. Reset the overrun bit in the adapter status register.
   k. Read the adapter status register to check that the overrun bit has been correctly reset; if not reset, display E50.

11. Test for invalid character transmission.

   a. Set the WRAP mode in the modem control register.
   b. Load the transmit buffer with an invalid character (X'00').
   c. Set the receive and transmit mode in the adapter control register.
   d. Software loop to wait interrupt.
   e. Read the basic status to check for output request, input request, exception request, interrupt request; if not all present, display E50.
   f. Read the adapter status register to check for the invalid character bit; if not present, display E50.
   g. Reset the invalid character bit in the adapter status register.
   h. Read the adapter status register to check for correct reset; if not, display E50.

12. Break test.

   a. Set the WRAP mode in the modem control register.
   b. Set the receive and break bits in the adapter control register.
   c. Software loop to wait for interrupt.

d. Read the basic status register to check for exception request and interrupt request; if not present, display E50.
e. Read the adapter status register to check for the break byte detected bit; if not present, display E50.
f. Reset the break byte detected bit in the adapter status register.
g. Read the adapter status register to check for correct reset; if not reset, display E50.

IML STEP F07

CCA INTERNAL WRAP TEST MODE

1. Reset the CCA adapter.
2. Send a 200ms timer, and wait for the interrupt.
3. Check that it is a timer interrupt; if not, display E50.
4. Send a 25-second control timer.
5. Write the modem control register with DTR, RTS, and WRAP.
6. On interrupt, check that it is a modem interrupt; if not, display E50.
7. Disable the modem interrupt in the modem control register.
8. Set the adapter control register with a character length equal to 8 bits.
9. Load the transmit buffer with the character X'FE' (odd pattern).
10. Set adapter control register with transmit and receive modes.
11. On interrupt, check that it is an output request; if not, display E50.
12. Prepare the next character to send, decrement the previous character pattern until an odd pattern is found. If an all '0' pattern is reached, reset the transmit mode in the adapter control register, and go to step 14.
13. Load the transmit buffer with the previously found character.
14. Send a 1-second survey timer.
15. On interrupt, check that it is an input request; if not, display E50.
16. If the received character is not the last one, send a 1-second survey timer, then go to step 11.
17. If it is the last character to be received, reset the receive mode in the adapter control register.
18. Set the character length and load the transmit buffer with the first character to send, using successively:

    a. 7-bit character length with first character = X'7F'.
    b. 6-bit character length with first character = X'3E'.
    c. 5-bit character length with first character = X'1F'.

19. Go to step 10 until all character lengths have been tested.
20. End of CCA internal wrap test mode.

CONSOLE LINK TEST

This routine is entered only if a specific IML is requested; once entered, this routine loops until another IML is requested.

To run this test, unplug the cable of the console under test at the entry to the 3727, then insert in place of the 3727 the connector P/N 2667737.

1. Read the modem status register and the modem control register. If they are not 0 under mask (respectively X'DF' and X'F7') send the E53/E54 'console link test not possible on local/alternate console' display, depending on the console under test.
2. Reset the CCA adapter.
3. Send a 200ms timer.
4. On interrupt check that it is a timer interrupt; if not, display KO.
5. Send a 25-second control timer.
6. Write the modem control register with DTR, RTS and DCE local test wire.
7. Disable the modem interrupt in the modem control register.
8. Set the adapter control register with a character length equal to 8 bits.
9. Load the transmit buffer with the character X'FE' (odd pattern).
10. Set the adapter control register with transmit and receive modes.
11. On interrupt check that it is an output request; if not, display KO.
12. Prepare the next character to be sent, decrement the previous character pattern until an odd bit number pattern is found. If an all '0's pattern is reached, reset the transmit mode in the adapter control register, then go to step 14.
13. Load the transmit buffer with the previously found character.
14. Send a 1-second survey timer.
15. On interrupt check that it is an input request; if not, display KO.
16. Read the received character, check that it is as expected; if not, display KO.
17. If the received character is not the last one, send a 1-second survey timer, then go to step 11.
18. If it is the last character to be received, reset the receive mode in the adapter control register.
19. Set the character length and load the transmit buffer with the first character to send, using successively:

    a. 7-bit character length with first character = X'7F'.
    b. 6-bit character length with first character = X'3E'.
    c. 5-bit character length with first character = X'1F'.

20. Go to step 10 until all character lengths have been tested.
21. Send test OK display.
22. Go to step 2.

Hex Display	Code Description
E53	Console link test not possible on local console
E54	Console link test not possible on alternate console
E55	Console link test OK on local console
E56	Console link test OK on alternate console
E57	Console link test KO on local console
E58	Console test KO on alternate console

CONSOLE ATTACHED TEST

1. Read the modem status register.
2. Check if data set ready (DSR) is on; if yes, this is the end of the current test.
3. If not on, send an error display depending on the console which is connected.
4. If diagnostic IML is requested, go to step 3.
5. If not send a 500ms timer, then end current test.

Hex Display	Code Description
E51	Local console power off or not connected
E52	Alternate console power off or not connected

HEXADECIMAL DISPLAY TEST

This test is a visual check only to check that the hexadecimal displays themselves are good.

1. Set the hexadecimal display with the first value of the following series:
   - 000, 001, ... 00F (16 values).
   - 011, 022, ... 0FF (15 values).
   - 111, 222, ... FFF (15 values).
2. Wait 500 milliseconds.
3. Set hexadecimal display with blank value.
4. Wait 500 milliseconds.
5. Check if the end of the display table has been reached (FFF); if not, return to step 1 with the next display value.
6. End of hexadecimal display test.

IML STEP F08

Note: if automatic IML and the CCU is running, skip the MCC Test Parts 1, 2, and 3.

MCC TEST PART 1: BASIC ADAPTER TEST

1. Read status register 1, ignore pending interrupts, check only if the MOSS INOP bit is on; if not, display E20.
2. Read status register 2.
3. If the CCU clock check bit is on, display E21.
4. If the ACLK check bit is on, display E22.
5. If the CCU interconnect time out, the MIOC interconnect parity check, or the CCU parity check is on, display E23.
6. If any other error bit is present, display E20.
7. If the IML request is not of the power on type, not the start type, go to step 13.
8. If the IML request is of the start type, go to step 11.
9. If the IML request is of the power on type, write the status register with an all '0' pattern.
10. Read status register 1 and check that the MOSS INOP bit is off; if not, display E20.
11. Send a reset panel adapter command.
12. Read status register 1 and check that the MOSS INOP bit is on; if not, display E20.
13. End of MCC test part 1.

MCC TEST PART 2

Entry is at level 7.

1. Reset the master mask to avoid swapping PSWs.
2. Write the interrupt test bit in status register 2.
3. Read status register 2 and check if the expected data is present; if not, display E28.
4. Write status register 1 with the pattern X'F7'.
5. Read status register 1 and compare it with the data sent; if not equal, display E25 (any timer interrupt is ignored at this level).
6. Reset status register 1 except for the MOSS NOP bit.
7. Reset status register 2.
8. Read status register 1 and check if it is as expected; if not, display E25.
9. Enable the master mask to allow the interrupt test.
10. Send an invalid command (X'0F') on the MCC adapter.
11. Read the status register and check that it has not changed; if it has changed, display E25.
12. Write the interrupt test bit in status register 2.
13. Enable the CCU interrupt and set a CCU high level interrupt in status register 1.

14. During the level 1 treatment, reset the level 1 bit.

15. Check if a level 1 interrupt has occurred; if not, display E26.

16. Set the scanner interrupt bit in status register 1.

17. Check if a level 4 interrupt has occurred; if yes, display E25.

18. Enable the scanner interrupt in status register 1.

19. During the level 4 treatment, reset the enable scanner interrupt and the scanner interrupt bit in status register 1.

20. Check if a level 4 interrupt has occurred; if not, display E26.

21. Set a CCU low level interrupt and enable CCU interrupt in status register 1.

22. During the level 4 treatment, reset the CCU low level interrupt bit and the CCU interrupt bit in status register 1.

23. Check if a level 4 interrupt has occurred; if not, display E26.

24. Reset the interrupt test bit in status register 2.

25. Enable the timer interrupt bit in status register 1.

26. Start a software loop longer than 100 milliseconds.

27. During the level 1 treatment, reset the timer interrupt bit and enable timer interrupt in status register 1.

28. Check if a level 1 interrupt has occurred; if not, display E27.

29. Check the 100 milliseconds timer interrupt between 80 milliseconds and 120 milliseconds; if no interrupt, display E27.

30. Reset the timer interrupt bit and the enable timer bit in status register 1.

31. Reset status register 2, and check for correct reset; if incorrect reset, display E25.

32. End of MCC Test Part 2.

MCC TEST PART 3

1. Check if the requested IML is of the diagnostic type; if yes, go to end of MCC Test Part 3.

2. Check the CCU software status; if running or loaded go to end of MCC Test Part 3.

3. Swap PSW to run this test on level 0.

4. Send a reset panel adapter command.

5. Send a read direct command from address 0.

6. Software timer to wait for a CCU time out.

7. Read status register 2 and check the CCU time out bit and the adapter check; if not present, display E29.

8. Reset the panel adapter.

9. Read status register 2 and check the correct reset of the CCU time out bit and the adapter check bit; if not reset, display E29.

10. Reset the MOSS INOP bit in status register 1.

11. Set the test interrupt bit in status register 2.

12. Send a read direct command from address 0.

13. Read status register 2 and check that the test interrupt, adapter check, MIOC parity, and CCU parity bits are on; if not all present, display E29.

14. Send a reset panel adapter command.

15. Read status register 2 and check the reset of the bits previously set; if not correctly reset, display E29.

16. Reset the MOSS INOP bit in status register 1.

17. Set the test interrupt bit in status register 2.

18. Send a write direct command to address 0.

19. Read status register 2 and check that the test interrupt, adapter check and CCU parity bits are all on; if not all present, display E29.

20. Send a reset panel command.

21. Read status register 2 and check the reset of the bits previously set; if not reset, display E29.

22. Check that there is no level 0 request in the IOIRR; if there is, display E29.

23. End of MCC Test Part 3.

MMC TEST PART 1

This test runs on level 0.

1. Read the storage expansion status register.

2. If any MCPC bit is on, display E40.

3. If 16K block installed, display E44.

4. If the inhibit storage expansion is not on, display E41.

5. If the other bits not zero, display E41.

6. Read each TTA location, then write each TTA location with its own TTA address.

7. If no parity error occurs after the complete TTA array read, go to step 10.

8. If a TTA parity error occurs, check if a level 0 has been requested in the IOIRR; if not, display E41.

9. Reset the TTA parity error bit in the storage expansion status register, and check for a correct reset; if reset not done, display E41.

10. Set the inhibit bit in the storage expansion control register, then check inhibit bit in the storage expansion status register; if not set, display E41.

11. Set the enable bit in the storage expansion control register, then check the enable bit in the storage expansion status register; if not set, display E41.

12. Set the reset inhibit bit in the storage expansion control register, then check the inhibit bit in the storage expansion status register; if not reset, display E41.

13. Scan up the entire TTA array to write X'FF' in each position.

14. Scan down the entire TTA array, read each location to check that there is no TTA parity error bit in the storage expansion status register; if on, display E41.

15. Check the received pattern; if not equal to X'FF', display E41.

16. During the scan down, store the pattern X'80' in each TTA array location.

17. Scan up the entire TTA array, read each location to check that there is no TTA parity error bit in the storage expansion status register; if the bit is on, display E41.

18. Check the received pattern; if not equal to X'80', display E41.

19. During this scan up, store the pattern X'00' in each TTA array location.

20. Scan down the entire TTA array, and read each location to check that there is no TTA parity error bit in the storage expansion status register; if the bit is on, display E41.

21. Check the received pattern; if not equal to X'00', display E41.

22. Using the TTA array reserved for level 0, check if there is a pluggable module installed in the correct place.

    a. If no module is installed, display E43

    b. If the module is installed at the wrong place, display E42.

23. Set TTA addresses X'FC11' through X'FC1F' with the values X'10' through X'17' to prepare for the 32K pluggable module test on level 0.

24. Scan up the entire 32K module, read each location, and check the received data; if not equal to X'FF', or if any MCPC bit or IOIRR bit 0 is on, display E31.

    If the read is OK, write X'80' in each location.

25. Scan up the entire 32K module, read each location, and check the received data; if not equal to X'80', or if any MCPC bit or IOIRR bit 0 is on, display E31.

    If the read is OK, write X'00' in each location.

26. Scan down the entire 32K module, read each location, and check the received data; if not equal to X'00', or if any MCPC bit or IOIRR bit 0 is on, display E31.

27. Disable and set inhibit in the storage expansion control register.

28. End of Memory Expansion Part 1.

IML STEP F09

MMC TEST PART 2

1. Set enable and reset inhibit in the storage expansion control register.

2. Prepare the Translate Table array for the interrupt addressing test. For each interrupt level, set block 2 to the successive values:

   ```
   Level 0  --->  2
   Level 1  --->  3
   Level 2  --->  4
   Level 3  --->  5
   Level 4  --->  6
   Level 5  --->  7
   Level 6  --->  8
   Level 7  --->  9
   ```

3. Initialize the PSW's for the interrupt addressing test.

4. Initialize main storage with the test patterns for the addressing test.

   ```
   Level 0  real address  X'2FFF'  pattern  80
   Level 1  real address  X'3FFF'  pattern  40
   Level 2  real address  X'4FFF'  pattern  20
   Level 3  real address  X'5FFF'  pattern  10
   Level 4  real address  X'6FFF'  pattern  08
   Level 5  real address  X'7FFF'  pattern  04
   Level 6  real address  X'8FFF'  pattern  02
   Level 7  real address  X'9FFF'  pattern  01
   ```

5. Interrupt mechanism test. Run the same test for each level from 0 to 7. Read the virtual address X'2FFF' and check the read pattern with the expected pre-set value; if not equal, display E41.

6. When all levels have been tested, return to level 0.

7. Write the protect bit in the translate table array level 0 block 2.

8. Try to write in main storage, the virtual address X'2FFF'.

9. Read the storage expansion status register and check for the presence of the storage protect violation bit, if not present display E41.

10. Read the IOIRR register and check for a level 0 request; if no level 0 request, display E41.

11. Reset the write protect violation in the storage expansion control register.

12. Read the storage expansion status register and check for the correct reset of the write protect violation; if not reset, display E41.

13. Read the main storage virtual location 2FFF and check it is as was done before the aborted write; if not, send error display E41.

14. Disable the storage expansion feature in the storage expansion control register.

15. End of MMC test part 2.

MOVE BUFFER CONTROL LIST

Move the buffer control list into the operational program communication area.

MOVE MOSS LOADER PROGRAM

Move the MOSS loader program to the end of storage (address X'17000').

IML STEP F0A

This is the end of the IML Diagnostic Part; pass control to the MOSS loading program.

Table of Contents for Chapter 7

CHAPTER 7. TRSS DIAGNOSTICS

The token ring subsystem (TRSS) diagnostic group consists of one IFT (T) that tests the TRM (Token ring multiplexor) and TIC (token ring interface) cards that are present on the token ring adapter (TRA).

The token ring subsystem (TRSS) diagnostic group runs under the control of the DCM in the MOSS.

The diagnostic programs reside on the MOSS service diskette and are invoked and controlled from the MOSS console. While running diagnostics, the TRA is logically "disconnected" from the CCU and may only communicate with the MOSS.

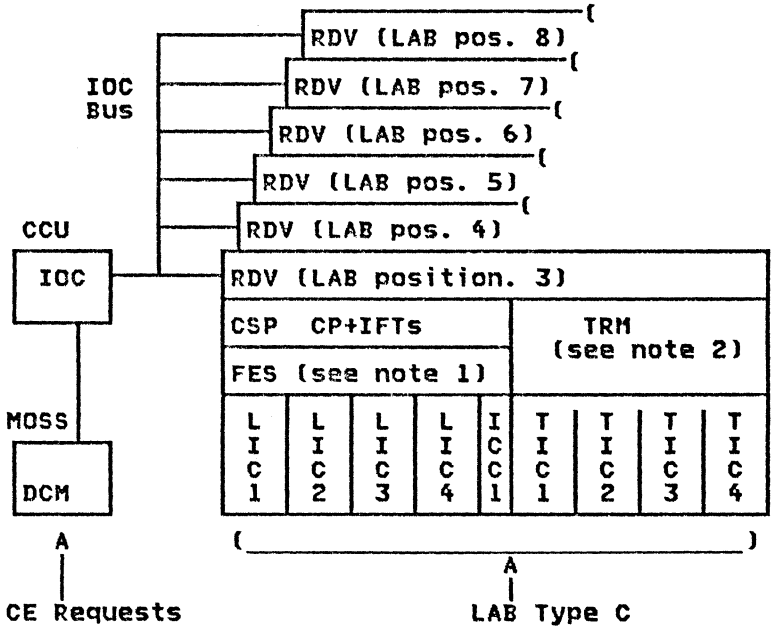


Figure 7-1. TRSS Diagnostics

Notes:

1. The interface between the FES and the LIC can be disabled by the diagnostic program to prevent LIC interference on the FES card.
2. The TRA must always be in the **second** (even) position on a LAB type C, a scanner being in the first position.

Before running the TRSS diagnostic group you must ensure that the CCU, IOC bus, and CA diagnostic groups work properly. If not, the results given by the TRSS diagnostic group may be of no value, or misleading.

SELECTION

For running off-line diagnostics, see Chapter 1.

The TRSS diagnostics have only one IFT (T), divided into eight sections (TA through TH) that can be loaded and executed one at a time.

Each section is divided into a set of routines. The shortest executable test is the routine.

The DCF provides the following diagnostic selection capabilities:

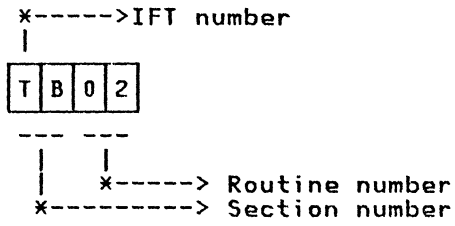
DIAG==>_

7	TRSS group selected
T	TRSS IFT selected
Tx	Specific section x in IFT T
Txyy	Specific routine yy in section Tx

For specific section and routine selection, see routine lists on following pages, and in Chapter 1.

Move the cursor from its initial position (DIAG==>) to the next after each parameter is entered. To skip a parameter entry, press the --> key.

To interpret correctly the results of a selected section or routine, make sure the preceding IFTs, sections, and routines in the group are running without error. The routine identification contains the IFT number, the section number, and the routine number as follows:



ADP#==>_ This field is used to enter the TRA number in the range 6 to 16 (only even numbers are valid). If no TRA is selected, the diagnostic will run on all TRAs defined in the Configuration Data File (CDF), in conjunction with the adapter request.

LINE==>_ This field is used to enter the ring attachment number (TIC 1, 2, 3, or 4). When no TIC is selected, the diagnostic will run on all TICs defined in the CDF (in conjunction with the adapter request).

OPT==> N For option display and description, see Chapter 1.

NUMBER OF RUNS PER REQUEST

The following table indicates how many times a section is run according to the selection request.

Select ADP#	Select LINE#	Number of Runs per Request
No	No	TA through TE: once per TRA TA and TC: once per TIC
Yes	No	As above for the selected TRA
Yes	Yes	TA through TE: once TA and TC: once on the selected TIC

Diagnostic Panel Sample

```
PROCESS  STOP-CCU-CHK SERVICE-MODE
          BYP-ADP-CHK

D:DIAGNOSTICS                                  SP:CCU STOP    Q:DATE/TIME
                                U:UTILITY PGM  ST:CCU START   T:TERMINATE
E:ERROR LOG                                    RS:CCU RESET

DIAG  |ADP# |LINE |
                  |
1 ALL             |
2 CCU             |
3 IOCB            |
4 CA  |1-> 6|     |
5 TSS |1->16|0->31|
6 OLT |1-> 6|     |
7 TRSS|6->16|1->4 |
                  |
       AND        |                                  DIAG - RUN INIT
                  |
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> TF    ADP#==> 6   LINE==> 1   OPT==> N
===>
```

On the above screen, section TF will run on TIC number 1.

Press SEND to execute the request.

Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

S T NOSTIC GROUP RUNNING TIME

The times below are related to a TRSS diagnostic group request (DIAG=7) for one specific TRA (ADP# = x).

Section	Time
CCU Init	10 seconds
Run Init	1 min 13s
TA	17 seconds
TB	14 seconds
TC	11 seconds
TD	10 seconds
TE	11 seconds
TF	2s + 2s/TIC
TG	2s + 8s/TIC
TH	2s + 12s/TIC

The diagnostic running time needed for the maximum TRSS configuration (2 TRAs with 8 TICs), not including initialization, will be 4 minutes, 20 seconds.

TRM TESTING (SECTIONS TA THROUGH TE)

The routines testing the TRM card are ordered so that the first routines test the simplest functions using the smallest amount of hardware. Later routines will then use the tested logic to test larger functions using additional hardware. The portion of the hardware that has been verified grows with each routine until the entire TRM has been tested. Though never used in normal operation, the functions provided through the Diagnostic Register must be tested since they will be used to verify the normal operational logic.

The following functional areas of the TRM will be tested in wrap mode. TIC actions/responses will be simulated by diagnostic timing logic.

Invalid PIO Detection

PIO commands will be issued with bad parity or invalid opcodes to test the ability of the TRM to recognize invalid IOHs.

TRM Registers/Data Buffer

The data buffer and registers of the TRM will be tested with selected sequences of patterns using the PIO write/read commands. Some errors will be generated to check the capability of TRM internal checkers.

CONNECT/DISCONNECT Operations

The STOP (DISCONNECT) and START (CONNECT) PIO commands are issued and the sequences of TRM actions and settings of the Level 1 Error Status Register are verified.

Programmed Reset

Data is put into each of the TRM registers, then a PROGRAMMED RESET is issued. The status of the registers is checked to verify the proper action on a reset.

MMIO Operations

Signals will be generated by diagnostic logic in the TRM to simulate TIC responses in MMIO. This checks the start of MMIO operational timing as well as the wrap mode.

Error Detection

The hardware checkers are tested by using the Diagnostic Register to force parity errors, idle state errors, and interface time outs. Detection, reporting, and logging of the errors will be checked.

TIC Interrupts

TIC interrupts are simulated using the Interrupt Request register of the TRM. The interrupt reporting (INTERRUPT TO MOSS), logging of type into the appropriate Level 2 Error Status Register, LID calculation mechanism, IR scan wheel, and Inhibit Interrupt functions are all tested.

DMA/Cycle Steal Operations

DMA/CS operations with TIC will be simulated using the TRM's Bus Request register. Both modes of transfer (EVEN CCU and ODD CCU) will be simulated for a byte count given in the Diagnostic Register. The swapping mechanism will be tested for transfers to odd CCU addresses. The BR scan wheel and Inhibit DMA functions are also verified.

DMA Error Management

Errors are forced at different times during a DMA operation and the proper completion of the operation (CSCW change, valid pattern, etc.) is verified.

MASK Function

The MASK PIO command is issued to the TRM in CONNECT mode, and the masking of all interrupts except the one generated by the end of the DISCONNECT operation is verified. The reset of the MASK function by a PROGRAMMED RESET is also verified.

TIC TESTING (SECTIONS TF AND TG)

Each TIC routine will be run on only one TIC at a time with the remaining TICs frozen. It is assumed that the TRM is fully operational or has already been tested before the TIC routines are run.

TIC Reset and Initialization

A TIC is reset and the results of its internal tests are obtained. The initialization procedure involves the MMIO and DMA functions and will be performed after the reset to verify these operations.

TIC Lobe Test/Interrupt Generation

The TIC internal Lobe Test is run by opening the TIC and the results of the test are obtained. Communication with the TIC through DMA operations to/from the SCB and SSB and through TIC-to-system interrupts is tested.

Non-Wrap DMA Errors

The handling of errors on the TIC-TRM interface (which cannot be tested in WRAP mode) is verified here. Specifically, the BERR line to the TIC, degating of TIC interrupts, DMA retry, and Adapter Check interrupts are tested.

DIAGNOSTIC DESCRIPTIONS BY ROUTINE

This section consists of a detailed account of each of the TRSS diagnostic routines. Each routine is described in three sections:

1. The **Description** section gives the function area being tested and the method used to test it.

2. The **Commands/Functions Covered** section lists the commands and functions used (and thereby tested) in this routine that have not been covered by previous routines.

3. The **ERC Description** lists and describes the various error reference codes that may appear in each routine.

,escription

This routine initializes the TRA to prepare it for the running of diagnostic routines. It tests the MIOH commands that will be issued to initialize the TRM hardware before all subsequent routines.

First, a GET TRM CONTROL REGISTER command is issued to allow communication with the TRA in case the diagnostics were called immediately after a POR or a programmed reset. Then the RESET TRM bit will be set to '0' so it may be used as an indicator later.

The WRAP MODE and PIO/DMA bits in the Diagnostic Register are set to '1' while the START and remaining bits are set to '0'. (Reset of the WRAP MODE bit will be included in TIC routines.)

The RESET TIC bits in the TIC Control Register are set to '1' to put any installed TIC in the reset state.

Finally, the IR/BR register is cleared to prevent unexpected interrupts from occurring when the START bit is used in later tests.

Commands/Functions Covered:

- SET/GET TRM CONTROL REGISTER
- WRITE/READ DIAGNOSTIC REGISTER
- SET/GET TIC CONTROL REGISTER
- WRITE/READ IR/BR REGISTER

ERC Description

Routine ERC	Meaning	FRU List
TA01	SETUP	TRM
001x	Interrupt on GET TRM Control register	
002x	Interrupt on SET TRM Control register	
003x	Interrupt on WRITE DIAG register	
004x	Interrupt on READ DIAG register	
0050	Read data not equal to write data	
006x	Interrupt on SET TIC Control	
007x	Interrupt on GET TIC Control	
0080	Read data not equal to write data	
009x	Interrupt on WRITE IR/BR register	
00Ax	Interrupt on READ IR/BR register	
00B0	Read data not equal to write data	

TA02: INVALID PIO DETECTION

Description

This routine tests the ability of the TRM to detect invalid IOH commands (unassigned IOH codes) and parity errors in address and command data (TA time data).

To test for the detection of parity errors, a PIO WRITE DIAGNOSTIC REGISTER will be issued to set up the diagnostic logic to force an error at TA time. Next, a valid PIO opcode will be issued. The TRM should not respond to this PIO, and this will be indicated by a time out on the IOC bus. The Level 1 Error Status Register should indicate an I/O CHECK.

Next, each unassigned PIO opcode will be issued. The IOC bus should time out and the error should be logged in the Level 1 Error Status Register as an invalid IOH.

Commands/Functions Covered:

- INVALID PIO checker
- Parity checker to IOC bus
- Diagnostic ability to force errors at TA time
- Reset of START bit in Diagnostic Register
- GET L1 ERROR STATUS command
- Logging of IOH INVALID and I/O CHECK in Level 1 Error Status Register

ERC Description

Routine ERC	Meaning	FRU List
TA02	INVALID PIO DETECTION	TRM
	TA Bad Parity on Byte 0	
0010	Expected timeout not received	
002x	Unexpected interrupt on GET L1 register	
0030	Improper setting, Level 1 register	
0040	Improper setting, Diag register	
	TA Bad Parity on Byte 1	
0050	Expected timeout not received	
006x	Unexpected interrupt on GET L1 register	
0070	Improper setting, Level 1 register	
0080	Improper setting, Diag register	
	TA Bad Parity on Both Bytes	
0090	Expected timeout not received	
00Ax	Unexpected interrupt on GET L1 register	
00B0	Improper setting, Level 1 register	
00C0	Improper setting, Diag register	
	Invalid IOH Opcode	
00D0	Expected timeout not received	
00E0	Improper setting, Level 1 register	

TA03: TRM CONTROL REGISTER

Description

This routine tests the two-bit TRM Control Register with selected patterns of bits using the SET/GET TRM CONTROL REGISTER command.

Commands/Functions Covered:

- TRM Control Register

ERC Description

Routine ERC	Meaning	FRU List
TA03	TRM CONTROL REGISTER	TRM
001x	Interrupt on SET TRM Control	
002x	Interrupt on GET TRM Control	
0030	Read data not equal to write data	

TA04: TIC CONTROL REGISTER

Description

This routine tests the TIC Control Register with selected patterns of bits using the SET/GET TIC CONTROL REGISTER command.

Commands/Functions Covered:

- TIC Control Register

ERC Description

Routine ERC	Meaning	FRU List
TA04	TIC CONTROL REGISTER	TRM
001x	Interrupt on SET FL Control	
002x	Interrupt on GET FL Control	
0030	Read data not equal to write data	

TA05: TRM DATA BUFFER

Description

The TRM data buffer and extended buffer are tested with selected patterns of bits.

Commands/Functions Covered:

- READ/WRITE BUFFER REGISTER
- READ/WRITE EXTENDED BUFFER REGISTER

ERC Description

Routine ERC	Meaning	FRU List
TA05	TRM DATA BUFFER	TRM
001x	Interrupt on WRITE BUFFER register	
002x	Interrupt on READ BUFFER register	
0030	Read data not equal to write data (buf)	
004x	Interrupt on WRITE EX BUF register	
005x	Interrupt on READ EX BUF register	
0060	Read data not equal to write data (ex)	

TA06: IR/BR REGISTER

Description

The IR/BR register is tested with selected patterns using the WRITE IR/BR and READ IR/BR PIO commands. The TRM must be in wrap mode. In wrap mode, the DMA and TIC interrupt operations are initiated with the START bit in the Diagnostic Register, hence we are able to set the bits in the IR/BR register without generating any interrupt or DMA operations.

Commands/Functions Covered:

- IR/BR REGISTER

ERC Description

Routine ERC	Meaning	FRU List
TA06	IR/BR REGISTER	TRM
001x	Interrupt on WRITE IR/BR	
002x	Interrupt on READ IR/BR	
0030	Read data not equal to write data	

LID BUFFER

Description

The LID Buffer is tested with selected patterns using the LOAD LID BASE and READ LID BASE PIO commands.

Commands/Functions Covered:

- LOAD LID BASE
- READ LID BASE

ERC Description

Routine ERC	Meaning	FRU List
TA07	LID BUFFER 001x Interrupt on LOAD LID BASE 002x Interrupt on READ LID BASE 0030 Read data not equal to write data	TRM

TA08: DIAGNOSTIC REGISTER

Description

This routine tests the Diagnostic Register with selected patterns using the WRITE and READ Diagnostic Register commands.

The register is tested first with a series of patterns that have a '0' in the START bit position. This prevents the diagnostic logic from forcing any errors that are specified in the remaining bits of the register. Then the START bit position is tested by setting the other bits so that no error is specified and turning the START bit on and off.

Commands/Functions Covered:

- Diagnostic Register

ERC Description

Routine ERC	Meaning	FRU List
TA08	DIAGNOSTIC REGISTER 001x Interrupt on WRITE DIAG register 002x Interrupt on READ DIAG register 0030 Read data not equal to write data	TRM

TA09: PROGRAMMED RESET

Description

This routine tests the programmed reset function of the TRM. All writeable registers are initialized and a programmed reset is issued. The contents of all registers are then checked to verify the function of the reset.

Commands/Functions Covered:

- PROGRAMMED RESET TRM command
- Programmed reset function
- Reset of "RESET latch" by GET TRM CONTROL register command
- GET MOSS ERROR STATUS REGISTER command
- GET L2 ERROR STATUS commands

ERC Description

Routine ERC	Meaning	FRU List
TA09	PROGRAMMED RESET	TRM
001x	Unexpected interrupt during setup	
002x	Interrupt on PROGRAMMED RESET	
003x	Unexpected interrupt, GET TRM Control	
0040	Improper setting, TRM Control register	
005x	Interrupt on GET L1 ERR STAT	
0060	Improper setting, L1 ERR STAT	
0070	Diagnostic register not cleared	
0080	Wrong setting, TIC Control register	
0090	IR/BR not cleared	
00A0	Data Buffer not cleared	
00B0	Extended Data Buffer not cleared	
00C0	LID Base register not cleared	
00Dx	Interrupt during register reads	
00Ex	Unexpected interrupt, GET L2 STAT 1	
00F0	TIC 1 L2 Error Status register not cleared	
010x	Unexpected interrupt, GET L2 STAT 2	
0110	TIC 2 L2 Error Status register not cleared	
012x	Unexpected interrupt, GET L2 STAT 3	
0130	TIC 3 L2 Error Status register not cleared	
014x	Unexpected interrupt, GET L2 STAT 4	
0150	TIC 4 L2 Error Status register not cleared	
016x	Interrupt on GET MOSS STATUS	
0170	MOSS Error Status register not '0's	

TA0A CYCLE STEAL CONTROL WORD

Description

The CSCW is read to check the pattern being sent to the CCU in DMA operations (since it is not available during a DMA operation). READ CSCW is a diagnostic command which returns the CSCW sent when errors have been detected in the CCU address (short/direct).

Commands/Functions Covered:

- CSCW short/direct pattern
- READ CSCW command

ERC Description

Routine ERC	Meaning	FRU List
TA0A	CYCLE STEAL CONTROL WORD	TRM
001x	Unexpected interrupt, READ CSCW	
0020	Improper CSCW pattern	

TB01: WRAP MODE (MMIO)

Description

This routine tests the ability of the diagnostic logic to simulate the TIC timing and data transfers in "wrap mode" for MMIO operations.

By using different combinations of TIC numbers and TIC registers, the mapping of PIO to MMIO can be verified. (The CS, RS, and RNW outputs of the TRM cannot be verified in wrap mode; only the start of the MMIO operational timing is verified here.)

Valid MMIO writes are issued, each followed by an MMIO read. No interrupts or time outs are expected, and the data received in each read operation should be that transmitted in the previous write operation. (Read data is wrapped from the TRM data buffer.)

Commands/Functions Covered:

- PIO/MMIO mapping
- Diagnostic wrap function
- Diagnostic MMIO timing

ERC Description

Routine ERC	Meaning	FRU List
TB01	WRAP MODE (MMIO)	TRM
001x	Unexpected interrupt, Write DATA, TIC 1	
002x	Unexpected interrupt, Read DATA, TIC 1	
0030	Inconsistent data, Write not equal to Read	
004x	Unexpected interrupt, Write DATA+, FL2	
005x	Unexpected interrupt, Read DATA+, FL2	
0030	Inconsistent data, Write not equal to Read	
007x	Unexpected interrupt, Write ADDRESS, TIC 3	
008x	Unexpected interrupt, Read ADDRESS, TIC 3	
0090	Inconsistent data, Write not equal to Read	
00Ax	Unexpected interrupt, Write Interrupt, TIC 4	
00Bx	Unexpected interrupt, Read Interrupt, TIC 4	
00C0	Inconsistent data, Write not equal to Read	

TB02: TD BAD PARITY

Description

This routine tests the ability of the TRM to force and detect errors at the IOC bus interface (checker #1). The proper logging of the Level 1 Error Status Register is also checked.

A TRM internal register is initialized with data and the Diagnostic Register is set up to force a parity error at TD time on byte 0. A PIO WRITE to the internal register is then issued and the instruction should time out. The reset of the START bit and the logging of the error in the L1 Error Status Register is then verified. The internal register is read and should remain as initialized; the data sent in the PIO WRITE should not have been placed in the register. This is repeated for B1 and both B0 and B1 simultaneously.

The TD Bad Parity error is set up again and a PIO READ is issued. The IOC should receive bad parity and cause the TRM to log the L1 Error Status. The reset of the START bit and the L1 Error Status Register setting are checked as before.

This portion of the test is performed for errors on B0, B1 and on both bytes simultaneously.

Commands/Functions Covered:

- Forcing of TD error to IOC bus interface
- IOC bus interface parity checker (#1)

ERC Description

Routine ERC	Meaning	FRU List
TB02	TD BAD PARITY	TRM
	Error Forced During Write, B0	
0010	No time out	
002x	Unexpected interrupt, GET L1 PIO	
0030	Wrong setting, L1 Error Status	
0040	START bit not reset, Diagnostic register	
0050	Register contents changed	
	Error Forced During Write, B1	
0060	No time out	
007x	Unexpected interrupt, GET L1 PIO	
0080	Wrong setting, L1 Error Status	
0090	START bit not reset, Diagnostic register	
00A0	Register contents changed	
	Error Forced on Write, B0+B1	
00B0	No time out	
00Cx	Unexpected interrupt, GET L1 PIO	
00D0	Wrong setting, L1 Error Status	
00E0	START bit not reset, Diagnostic register	
00F0	Register contents changed	
	Error Forced on Read, B0	
0100	No L1 for parity error, IOC Bus	
011x	Unexpected interrupt, GET L1 PIO	
0120	Wrong setting, L1 Error Status	
0130	START bit not reset, Diagnostic register	
	Error Forced on Read, B1	
0140	No L1 for parity error, IOC Bus	
015x	Unexpected interrupt, GET L1 PIO	
0160	Wrong setting, L1 Error Status	
0170	START bit not reset, Diagnostic register	
	Error Forced on Read, B0 + B1	
0180	No L1 for parity error, IOC Bus	
019x	Unexpected interrupt, GET L1 PIO	
01A0	Wrong setting, L1 Error Status	
01B0	START bit not reset, Diagnostic register	

TB03: BAD PARITY TO INTERNAL REGISTERS

Description

This routine tests the ability of the TRM to force parity errors to its internal registers using the Diagnostic Register. The hardware reset of the START bit and detection of a parity error by the IOC bus interface checker (#1) is also tested.

The Diagnostic Register is set up to force a parity error on byte 0 to an internal register (error is forced on the following command). A PIO WRITE is issued and no interrupt or time out should result (the parity error is not forced to a checker, it is stored in the register). The Diagnostic Register is then read to verify the reset of the START bit.

Next, a PIO READ is issued for the same register and should result in an IOC bus-in parity error because of the parity error stored in the register. The proper setting of bits in the Level 1 Error Status Register will be verified.

Only the following internal registers store data with parity. Writing to the other internal registers with bad parity has no effect.

DATA BUFFER
EXTENDED DATA BUFFER (byte 0 not implemented)
LID BASE REGISTER

The test is performed for the above three registers forcing errors on byte 0, byte 1, and B0 and B1 simultaneously.

Commands/Functions Covered:

- IOC bus interface parity checker (#1)
- Forcing of parity error to internal register
- Reset of START bit
- Logging of error in L1 Error Status Register

ERC Description

Routine ERC	Meaning	FRU List
TB03	BAD PARITY TO INTERNAL register	TRM
	Register = Data Buffer	
	Byte 0 Error	
001x	Interrupt on write to register	
0020	START bit not reset, Diagnostic register	
0030	No L1 for IOC bad parity	
0040	Wrong setting, L1 Error Status	
	Byte 1 Error	
005x	Interrupt on write to register	
0060	START bit not reset, Diagnostic register	
0070	No L1 for IOC bad parity	
0080	Wrong setting, L1 Error Status	
	Byte 0 and Byte 1	
009x	Interrupt on write to register	
00A0	START bit not reset, Diagnostic register	
00B0	No L1 for IOC bad parity	
00C0	Wrong setting, L1 Error Status	
	Register = Extended Data Buffer	
	Byte 1 Error	
00Dx	Interrupt on write to register	
00E0	START bit not reset, Diagnostic register	
00F0	No L1 for IOC bad parity	
0100	Wrong setting, L1 Error Status	
	Byte 0 and Byte 1	
011x	Interrupt on write to register	
0120	START bit not reset, Diagnostic register	
0130	No L1 for IOC bad parity	
0140	Wrong setting, L1 Error Status	
	Register = LID Base	
	Byte 0 Error	
015x	Interrupt on write to register	
0160	START bit not reset, Diagnostic register	
0170	No L1 for IOC bad parity	
0180	Wrong setting, L1 Error Status	
	Byte 1 Error	
019x	Interrupt on write to register	
01A0	START bit not reset, Diagnostic register	
01B0	No L1 for IOC bad parity	
01C0	Wrong setting, L1 Error Status	
	Byte 0 and Byte 1	
01Dx	Interrupt on write to register	
01E0	START bit not reset, Diagnostic register	
01F0	No L1 for IOC bad parity	
0200	Wrong setting, L1 Error Status	

TB04: INTERNAL BUS PARITY ERROR

Description

This routine tests the ability of the Diagnostic Register to force an error on the internal bus. The detection (by internal bus parity checker #3), reporting, and logging of the error are also checked.

The Diagnostic Register is set up to force a parity error on internal bus byte 0. An MMIO read is then issued and the correct reporting (INTERRUPT TO MOSS) and logging of the error is verified. This procedure is repeated for byte 1 then for both bytes 0 and 1.

Commands/Functions Covered:

- Forcing of internal bus parity error
- Internal bus parity checker (#3)
- Logging of error in L2 Error Status Register
- Interrupt to MOSS
- GET COMMAND COMPLETION
- LID Calculation (partial test)
- READ COMPUTED LID
- Reset of L2 Error Status Register after GET L2 ERROR STATUS command is received (partial--see TIC Interrupt test)

ERC Description

Routine ERC	Meaning	FRU List
TB04	INTERNAL BUS PARITY ERROR	TRM
	Error Forced on Byte 0	
0010	No L4 interrupt on MMIO read	
0020	Wrong data, command Completion	
0030	Incorrect LID value	
0040	START bit not reset, Diagnostic register	
0050	Improper setting, L2 Status register	
0060	L2 Status register not reset, second read	
	Error Forced on Byte 1	
0070	No L4 interrupt on MMIO read	
0080	Wrong data, command Completion	
0090	Incorrect LID value	
00A0	START bit not reset, Diagnostic register	
00B0	Improper setting, L2 Status register	
00C0	L2 Status register not reset, second read	
	Error Forced on Bytes 0 and 1	
00D0	No L4 interrupt on MMIO read	
00E0	Wrong data, command Completion	
00F0	Incorrect LID value	
0100	START bit not reset, Diagnostic register	
0110	Improper setting, L2 Status register	
0120	L2 Status register not reset, second read	

TB05: IDLE STATE ERROR ON SYSTEM BUS

Description

The ability of the TRM to force, detect, and properly report idle state errors at the TIC interface is tested.

The diagnostic register is set up to force an error to an idle state checker on the TIC interface. An MMIO is issued, and an interrupt to MOSS should result with the error properly logged in the appropriate Level 2 Error Status Register.

Two types of idle state errors (TRM internal and TIC interface type 2) will be forced to the idle state checker of TIC bus tags (checker #6). Only the TIC type 2 error will be forced to checker #7 (idle state checker of TIC bus) as this is the only error it is able to detect.

Commands/Functions Covered:

- Forcing of all types of idle state error
- Idle State checkers (#6 and #7)
- Logging of errors in L2 Error Status Register
- Reset of L2 Error Status Register after GET L2 ERROR STATUS command is received (partial--see TIC Interrupt test)

ERC Description

Routine ERC	Meaning	FRU List
TB05	IDLE STATE ERROR ON SYSTEM BUS	TRM
	Internal Error to Checker #6	
0010	No INTERRUPT TO MOSS	
0020	Wrong data, command Completion	
0030	Incorrect LID value	
0040	START bit not reset, Diagnostic register	
0050	Improper setting, Level 2 register	
0060	L2 Error Status register not reset	
	External Type 2 to Checker #6	
0070	No INTERRUPT TO MOSS	
0080	Wrong data, command Completion	
0090	Incorrect LID value	
00A0	START bit not reset, Diagnostic register	
00B0	Improper setting, Level 2 register	
00C0	L2 Error Status register not reset	
	External Type 2 to Checker #7	
00D0	No INTERRUPT TO MOSS	
00E0	Wrong data, command Completion	
00F0	Incorrect LID value	
0100	START bit not reset, Diagnostic register	
0110	Improper setting, Level 2 register	
0120	L2 Error Status register not reset	

TB06: DTACK TIME OUT (TIC BUS)

Description

This routine checks the ability of the TRM to simulate, detect, and report the failure of the TIC to send a DTACK response during an MMIO.

A late or missing DTACK response from a TIC is simulated by setting up the FORCE INTERFACE TIME OUT condition in the Diagnostic Register and then issuing an MMIO. A MOSS interrupt is expected and correct reporting and logging of the error is verified.

Note: The AS/DS timer will be tested in the "CSCW Change During DMA" test (AS) and the "Error Management During DMA" test (DS).

Commands/Functions Covered:

- Forcing of time out on TIC bus
- DTACK Timer (Checker #4)
- Logging of error in L2 Error Status Register

ERC Description

Routine ERC	Meaning	FRU List
TB06	DTACK TIME OUT (TIC BUS)	TRM
0010	No INTERRUPT TO MOSS	
0020	Wrong data, command Completion	
0030	Incorrect LID value	
0040	START bit not reset, Diagnostic register	
0050	Improper setting, Level 2 register	
0060	L2 Error Status register not reset	

TC01: TIC INTERRUPT

Description

This routine tests the TRM's ability to service interrupts from the TIC by setting the proper Level 2 Error Status Register bits and calculating a LID.

A value representing the TIC interrupt vector is placed into the TRM data buffer (the interrupt vector is taken from byte 1). The diagnostics simulate an interrupt request coming from a TIC by setting a bit in the IR register and the Diagnostic Register START bit. An interrupt to MOSS should be generated by the TRM. The interrupt vector is wrapped and its value is used to set the L2 Error Status Register. The setting of the appropriate Level 2 Error Status Register will be checked, and the LID will be read and verified.

This procedure will be repeated for three different TIC interrupt vectors (adapter check, SCB clear, any using LID A) for each of the TICs (initiated from each of the IR bit positions).

Commands/Functions Covered:

- Generation of interrupt to MOSS for IR in each TIC location
- Setting of Level 2 Error Status Register for TIC interrupt
- Reset of L2 Error Status Register after GET L2 ERROR STATUS command is received
- LID calculation

ERC Description

Routine ERC	Meaning	FRU List
TC01	TIC INTERRUPT	TRM
	SCB CLEAR Interrupt Vector, TIC 1	
0010	No INTERRUPT TO MOSS	
0020	Wrong data, command Completion	
0030	Incorrect LID value	
0040	START bit not reset, Diagnostic register	
0050	Improper setting, Level 2 register	
0060	L2 Error Status register not reset	
007x	Unexpected interrupt during test	
	ADAPTER CHECK interrupt Vector, FL1	
0080	No INTERRUPT TO MOSS	
0090	Wrong data, command Completion	
00A0	Incorrect LID value	
00B0	START bit not reset, Diagnostic register	
00C0	Improper setting, Level 2 register	
00D0	L2 Error Status register not reset	
00Ex	Unexpected interrupt during test	
	Vector For a Type A LID, TIC 1	
00F0	No INTERRUPT TO MOSS	
0100	Wrong data, command Completion	
0110	Incorrect LID value	
0120	START bit not reset, Diagnostic register	
0130	Improper setting, Level 2 register	
0140	L2 Error Status register not reset	
015x	Unexpected interrupt during test	
	SCB CLEAR Interrupt Vector, TIC 2	
0160	No INTERRUPT TO MOSS	
0170	Wrong data, command Completion	
0180	Incorrect LID value	
0190	START bit not reset, Diagnostic register	
01A0	Improper setting, Level 2 register	
01B0	L2 Error Status register not reset	
01Cx	Unexpected interrupt during test	
	ADAPTER CHECK Interrupt Vector, FL2	
01D0	No INTERRUPT TO MOSS	
01E0	Wrong data, command Completion	
01F0	Incorrect LID value	
0200	START bit not reset, Diagnostic register	
0210	Improper setting, Level 2 register	
0220	L2 Error Status register not reset	
023x	Unexpected interrupt during test	

Routine ERC	Meaning	FRU List
TC01	(continuation)	
	Vector For a Type A LID, TIC 2	
0240	No INTERRUPT TO MOSS	
0250	Wrong data, command Completion	
0260	Incorrect LID value	
0270	START bit not reset, Diagnostic register	
0280	Improper setting, Level 2 register	
0290	L2 Error Status register not reset	
02Ax	Unexpected interrupt during test	
	SCB CLEAR Interrupt Vector, TIC 3	
02B0	No INTERRUPT TO MOSS	
02C0	Wrong data, command completion	
02D0	Incorrect LID value	
02E0	START bit not reset, Diagnostic register	
02F0	Improper setting, Level 2 register	
0300	L2 Error Status register not reset	
031x	Unexpected interrupt during test	
	ADAPTER CHECK interrupt Vector, TIC 3	
0320	No INTERRUPT TO MOSS	
0330	Wrong data, command Completion	
0340	Incorrect LID value	
0350	START bit not reset, Diagnostic register	
0360	Improper setting, Level 2 register	
0370	L2 Error Status register not reset	
038x	Unexpected interrupt during test	
	Vector For a Type A LID, TIC 3	
0390	No INTERRUPT TO MOSS	
03A0	Wrong data, command Completion	
03B0	Incorrect LID value	
03C0	START bit not reset, Diagnostic register	
03D0	Improper setting, Level 2 register	
03E0	L2 Error Status register not reset	
03Fx	Unexpected interrupt during test	
	SCB CLEAR Interrupt Vector, TIC 4	
0400	No INTERRUPT TO MOSS	
0410	Wrong data, command Completion	
0420	Incorrect LID value	
0430	START bit not reset, Diagnostic register	
0440	Improper setting, Level 2 register	
0450	L2 Error Status register not reset	
046x	Unexpected interrupt during test	
	ADAPTER CHECK interrupt Vector, TIC 4	
0470	No INTERRUPT TO MOSS	
0480	Wrong data, command Completion	
0490	Incorrect LID value	
04A0	START bit not reset, Diagnostic register	
04B0	Improper setting, Level 2 register	
04C0	L2 Error Status register not reset	
04Dx	Unexpected interrupt during test	
	Vector for a Type A LID, TIC 4	
04E0	No INTERRUPT TO MOSS	
04F0	Wrong data, command Completion	
0500	Incorrect LID value	
0510	START bit not reset, Diagnostic register	
0520	Improper setting, Level 2 register	
0530	L2 Error Status register not reset	
054x	Unexpected interrupt during test	

TC02: ERROR MANAGEMENT DURING IACK

Description

This routine tests the operation of the TRM when errors are detected by the hardware checkers during the input of the TIC interrupt vector.

TIC interrupts are simulated as in the TIC Interrupt routine, but parity, timer, and idle state errors are forced during IACK via the Diagnostic Register. The TRM must detect the errors and properly log them (as errors detected by the TRM, not as TIC interrupts) in the Level 2 Error Status Register for the proper TIC. A type B LID should be obtained for a READ COMPUTED LID instruction.

The TIC Interrupt routine and the routines checking the internal bus parity checker, idle state checkers, and DTACK timer must be run prior to this routine.

Note: Parity is checked only on byte 1 of the interrupt vector.

Commands/Functions Covered:

- Logging of errors during IACK in L2 Error Status Registers
- Calculation of type B LID for error during IACK

ERC Description

Routine ERC	Meaning	FRU List
TC02	ERROR MANAGEMENT DURING IACK	TRM
	Internal Bus Bad Parity, Byte 1	
0010	No INTERRUPT TO MOSS	
0020	Wrong data, command Completion	
0030	Incorrect LID value	
0040	START bit not reset, Diagnostic register	
0050	Improper setting, Level 2 register	
0060	L2 Error Status register not reset	
007x	Unexpected interrupt during test	
	Internal Bus Bad Parity, Byte 0 and Byte 1	
0080	No INTERRUPT TO MOSS	
0090	Wrong data, command Completion	
00A0	Incorrect LID value	
00B0	START bit not reset, Diagnostic register	
00C0	Improper setting, Level 2 register	
00D0	L2 Error Status register not reset	
00Ex	Unexpected interrupt during test	
	Idle State Error, Checker #6 Internal	
00F0	No INTERRUPT TO MOSS	
0100	Wrong data, command Completion	
0110	Incorrect LID value	
0120	START bit not reset, Diagnostic register	
0130	Improper setting, Level 2 register	
0140	L2 Error Status register not reset	
015x	Unexpected interrupt during test	
	Idle State Error, Checker #6 Type 2	
0160	No INTERRUPT TO MOSS	
0170	Wrong data, command Completion	
0180	Incorrect LID value	
0190	START bit not reset, Diagnostic register	
01A0	Improper setting, Level 2 register	
01B0	L2 Error Status register not reset	
01Cx	Unexpected interrupt during test	
	Idle State Error, Checker #7 Type 2	
01D0	No INTERRUPT TO MOSS	
01E0	Wrong data, command Completion	
01F0	Incorrect LID value	
0200	START bit not reset, Diagnostic register	
0210	Improper setting, Level 2 register	
0220	L2 Error Status register not reset	
023x	Unexpected interrupt during test	
	System Bus Time Out	
0240	No INTERRUPT TO MOSS	
0250	Wrong data, command Completion	
0260	Incorrect LID value	
0270	START bit not reset, Diagnostic register	
0280	Improper setting, Level 2 register	
0290	L2 Error Status register not reset	
02Ax	Unexpected interrupt during test	

TC03: LEVEL 1 ERROR DURING READ COMPUTED LID (GLID)

Description

This routine tests the ability of the TRM to log the number of the TIC whose interrupt it is servicing into the L1 Error Status Register when an error (level 1) occurs on a READ COMPUTED LID command. The READ COMPUTED LID command has exactly the same function as the GET LID issued by the CCU.

A TIC interrupt is set and the Diagnostic Register is set up to force TA and TD parity errors during the READ COMPUTED LID command used to service the interrupt. The values used for the LID BASE and interrupt vector value are not important since the value of the LID will not be received. The Level 1 Error Status Register should indicate the number of the TIC whose interrupt is being serviced.

Commands/Functions Covered:

- Logging of TIC Number in L1 Error Status Register (READ LID BY MOSS)

ERC Description

Routine ERC	Meaning	FRU List
TC03	L1 ERROR IN READ LID BY MOSS	TRM
0010	Wrong data, command Completion	
0020	No L1 on READ LID	
0030	Wrong setting, L1 Error Status	

TC04: INHIBIT INTERRUPT

Description

The function of the INHIBIT INTERRUPT bits of the TIC Control Register is tested in this routine.

Bits are set in the IR register, then the INHIBIT INTERRUPT bits of the TIC Control Register are set. The IR bits should remain set after this command is issued, but a WRITE IR/BR should cause them to be reset.

Because this test is performed in WRAP MODE, no interrupt will be generated from the IR/BR (the START bit will not be set).

Commands/Functions Covered:

- Inhibit Interrupt function (of TIC Control Register)

ERC Description

Routine ERC	Meaning	FRU List
TC04	INHIBIT INTERRUPT	TRM
0010	Pending IR cleared by INH	
0020	INH failure on WRITE IR/BR	

TC05: IR SCAN WHEEL

Description

This routine tests the ability of the TRM to service TIC interrupts in the proper order. Bits are set in the IR register to represent TIC interrupt requests. The order in which the interrupts are serviced can be monitored by the LID value returned in a READ COMPUTED LID command.

The TIC Interrupt test must be run prior to this routine.

Commands/Functions Covered:

- IR Scan Wheel

ERC Description

Routine ERC	Meaning	FRU List
TC05	IR SCAN WHEEL	TRM
0010	No L4 interrupt, TIC 0 due	
0020	Wrong LID, TIC 0 not serviced	
0030	No L4 interrupt, TIC 2 due	
0040	Wrong LID, TIC 2 not serviced	
0050	No L4 interrupt, TIC 0 due	
0060	Wrong LID, TIC 0 not serviced	
0070	No L4 interrupt, TIC 1 due	
0080	Wrong LID, TIC 1 not serviced	
0090	No L4 interrupt, TIC 2 due	
00A0	Wrong LID, TIC 2 not serviced	
00B0	No L4 interrupt, TIC 3 due	
00C0	Wrong LID, TIC 3 not serviced	
00D0	No L4 interrupt, TIC 1 due	
00E0	Wrong LID, TIC 1 not serviced	

TC06: INHIBIT DMA

Description

The function of the INHIBIT DMA bits of the TIC Control Register is tested in this routine.

Bits are set in the BR register, then the INHIBIT DMA bits of the TIC Control Register are set. The BR bits should remain set after this command is issued, but a WRITE IR/BR should cause them to be reset.

Because this test is performed in WRAP MODE, no DMA will be generated from the IR/BR (the START bit will not be set).

Commands/Functions Covered:

- Inhibit DMA function (of TIC Control Register)

ERC Description

Routine ERC	Meaning	FRU List
TC06	INHIBIT DMA	TRM
0010	Pending BR cleared by INH	
0020	INH failure on WRITE IR/BR	

TC07: ERROR MANAGEMENT DURING GET L2 STATUS ERROR

This routine tests the ability of the TRM to disable the reset of the Level 2 Error Status registers when a level 1 error occurs during a GET L2 ERROR STATUS command. Normally, (when no level 1 error is detected) the Level 2 registers are reset by the GET L2 commands.

An internal bus parity error is forced during an MMIO with TIC 1. The generation of an interrupt to MOSS and the values returned for GET COMMAND COMPLETION and READ COMPUTED LID are verified. The Diagnostic Register is set up to force a TD parity error, and GET L2 ERROR STATUS is issued. The IOC should detect a parity error. The GET L2 command is issued again. The MMIO error should still be logged in the register. The register should not have been reset by the first GET L2 command issued.

The test is repeated with MMIOs to TICs 2-4 to test each of the Level 2 registers.

Commands/Functions Covered:

- L2 Error Status Register reset mechanism

ERC Description

Routine ERC	Meaning	FRU List
TC07	ERROR MANAGEMENT, GET L2 STATUS	TRM
	TIC 1	
0010	No L4 for internal bus bad parity	
0020	Wrong data, command Completion	
0030	Wrong LID value returned	
0040	No L1 on GET L2 Error Status	
0050	Wrong setting, L1 status register	
0060	L2 register has been reset	
	TIC 2	
0070	No L4 for internal bus bad parity	
0080	Wrong data, command Completion	
0090	Wrong LID value returned	
00A0	No L1 on GET L2 Error Status	
00B0	Wrong setting, L1 status register	
00C0	L2 register has been reset	
	TIC 3	
00D0	No L4 for internal bus bad parity	
00E0	Wrong data, command Completion	
00F0	Wrong LID value returned	
0100	No L1 on GET L2 Error Status	
0110	Wrong setting, L1 status register	
0120	L2 register has been reset	
	TIC 4	
0130	No L4 for internal bus bad parity	
0140	Wrong data, command Completion	
0150	Wrong LID value returned	
0160	No L1 on GET L2 Error Status	
0170	Wrong setting, L1 status register	
0180	L2 register has been reset	

TD01: DMA OPERATIONS

Description

This routine tests the ability of the TRM to process DMA operations. A DMA operation is simulated by the diagnostic hardware by indicating a byte count and the direction of the transfer in the Diagnostic Register. Data which will represent both the CCU address and the data to be transferred is written to the TRM data buffer and a bit is set in the BR register to simulate a TIC bus request.

Data transfers to both odd and even CCU starting addresses will be tested to check the TRM's swapping mechanism (used in transfers to odd addresses). These two modes (CCU ODD and CCU EVEN) are specified with the DATA MODE bit of the Diagnostic Register. The COUNT field of the Diagnostic Register is used to specify the number of bytes to be transferred. The operation is started with the Diagnostic Register START bit.

Commands/Functions Covered:

- Generation of CSR from BR in each TIC location
- Swapping mechanism for odd CCU starting addresses
- Diagnostic wrap for DMA (inbound data only)
- Diagnostic generation of DMA timing

ERC Description

Routine ERC	Meaning	FRU List
TD01	DMA OPERATIONS	TRM
	1 byte write, even address, TIC 0	
001x	Unexpected interrupt, DMA setup	
002x	Unexpected interrupt, end of DMA	
0030	Incorrect data transfer	
	2 byte write, even address, TIC 1	
004x	Unexpected interrupt, DMA setup	
005x	Unexpected interrupt, end of DMA	
0060	Incorrect data transfer	
	3 byte write, even address, TIC 2	
007x	Unexpected interrupt, DMA setup	
008x	Unexpected interrupt, end of DMA	
0090	Incorrect data transfer	
	4 byte write, even address, TIC 3	
00Ax	Unexpected interrupt, DMA setup	
00Bx	Unexpected interrupt, end of DMA	
00C0	Incorrect data transfer	
	2 byte read, even address, TIC 0	
00Dx	Unexpected interrupt, DMA setup	
00Ex	Unexpected interrupt, end of DMA	
00F0	Incorrect data transfer	
	1 byte write, odd address, TIC 0	
010x	Unexpected interrupt, DMA setup	
011x	Unexpected interrupt, end of DMA	
0120	Incorrect data transfer	
	4 byte write, odd address, TIC 0	
013x	Unexpected interrupt, DMA setup	
014x	Unexpected interrupt, end of DMA	
0150	Incorrect data transfer	
	2 byte read, odd address, TIC 0	
016x	Unexpected interrupt, DMA setup	
017x	Unexpected interrupt, end of DMA	
0180	Incorrect data transfer	
	4 byte read, odd address, TIC 0	
019x	Unexpected interrupt, DMA setup	
01Ax	Unexpected interrupt, end of DMA	
01B0	Incorrect data transfer	

TD02: CSCW CHANGE DURING DMA

Description

This routine tests the ability of the TRM to detect errors during the building of the CSCW and to change the CSCW from long/indirect to short/direct before sending it to the CCU.

A DMA operation is set up as in the previous test, but an error is forced on the data written to the TRM data buffer representing the CCU address. The TRM finds the parity error when the DMA operation is in progress and must change the CSCW to indicate the error and terminate the DMA without sending any data. A valid data pattern is sent to the CCU in place of the bad parity address. This prevents an IOC time out.

The test is repeated with an address with good parity but using the Diagnostic Register to force errors on the internal bus, idle state errors, and a time out on the TIC bus. Again, the TRM changes the CSCW and terminates the operation before sending the data.

The CSCW sent to the CCU cannot be read directly, so the status of the data area will be used to determine whether or not the appropriate action was taken with the data (hence whether or not the CSCW was changed).

The DMA Operations test must be run prior to this routine.

Commands/Functions Covered:

- CSCW change (from long/indirect to short/direct)
- AS/DS Timer (for missing AS)
- Sending of valid pattern in place of bad parity address
- Ability to end DMA operation before data transfer
- Logging of error in appropriate L2 Error Status Register
- Logging of all three error types for DMA in each L2 Error Status Register

ERC Description

Routine ERC	Meaning	FRU List
TD02	CSCW CHANGE DURING DMA Write, Bad Parity Address, TIC 0	TRM
001x	Unexpected interrupt, DMA setup	
0020	No L4 interrupt after DMA	
0030	Incorrect LID value	
0040	Improper setting, Level 2 register	
0050	CCU memory changed	
	TIC Bus Time Out, Type 1, TIC 0	
006x	Unexpected interrupt, DMA setup	
0070	No L4 interrupt after DMA	
0080	Incorrect LID value	
0090	Improper setting, Level 2 register	
00A0	CCU memory changed	
	Idle State, #6, Interrupt, TIC 1	
00Bx	Unexpected interrupt, DMA setup	
00C0	No L4 interrupt after DMA	
00D0	Incorrect LID value	
00E0	Improper setting, Level 2 register	
00F0	CCU memory changed	
	Idle State, #6, Type 2, TIC 0	
010x	Unexpected interrupt, DMA setup	
0110	No L4 interrupt after DMA	
0120	Incorrect LID value	
0130	Improper setting, Level 2 register	
0140	CCU memory changed	
	Idle State, #7, Type 2, TIC 1	
015x	Unexpected interrupt, DMA setup	
0160	No L4 interrupt after DMA	
0170	Incorrect LID value	
0180	Improper setting, Level 2 register	
0190	CCU memory changed	
	Internal Bus Parity, Interrupt, TIC 2	
01Ax	Unexpected interrupt, DMA setup	
01B0	No L4 interrupt after DMA	
01C0	Incorrect LID value	
01D0	Improper setting, Level 2 register	
01E0	CCU memory changed	
	TIC Bus Time Out, Type 1, TIC 1	
01Fx	Unexpected interrupt, DMA setup	
0200	No L4 interrupt after DMA	
0210	Incorrect LID value	
0220	Improper setting, Level 2 register	
0230	CCU memory changed	

Routine ERC	Meaning	FRU List
TD02	(continuation)	
	Idle State, #6, Type 2, TIC 2	
024x	Unexpected interrupt, DMA setup	
0250	No L4 interrupt after DMA	
0260	Incorrect LID value	
0270	Improper setting, Level 2 register	
0280	CCU memory changed	
	TIC Bus Time Out, Type 1, TIC 2	
029x	Unexpected interrupt, DMA setup	
02A0	No L4 interrupt after DMA	
02B0	Incorrect LID value	
02C0	Improper setting, Level 2 register	
02D0	CCU memory changed	
	Internal Bus Parity, Interrupt, TIC 3	
02Ex	Unexpected interrupt, DMA setup	
02F0	No L4 interrupt after DMA	
0300	Incorrect LID value	
0310	Improper setting, Level 2 register	
0320	CCU memory changed	
	TIC Bus Time Out, Type 1, TIC 3	
033x	Unexpected interrupt, DMA setup	
0340	No L4 interrupt after DMA	
0350	Incorrect LID value	
0360	Improper setting, Level 2 register	
0370	CCU memory changed	
	Idle State, #6, Type 2, TIC 3	
038x	Unexpected interrupt, DMA setup	
0390	No L4 interrupt after DMA	
03A0	Incorrect LID value	
03B0	Improper setting, Level 2 register	
03C0	CCU memory changed	
	Read, TIC Bus Time Out, Typ 1, TIC 0	
03Dx	Unexpected interrupt, DMA setup	
03E0	No L4 interrupt after DMA	
03F0	Incorrect LID value	
0400	Improper setting, Level 2 register	
0410	Data buffer changed	
0420	Extended data buffer changed	

TE01: ERROR MANAGEMENT DURING DMA

Description

This routine tests for the proper operation of the TRM when errors are detected by the hardware checkers during a DMA operation. Also tested in this routine is the Data Strobe checker.

DMA operations are set up as in the DMA Operations routine, but the Diagnostic Register is used to force parity and timer errors at specific points in the operation (specified in conjunction with the COUNT field). In wrap mode, the UDS/LDS and LAST XFER signals are generated by the diagnostic hardware according to the data mode and byte count specified in the Diagnostic Register. By writing an address that is inconsistent with the data mode specified (e.g. an even address in CCU ODD mode) to the TRM data buffer the diagnostic hardware can be made to generate invalid UDS/LDS combinations to the Data Strobe checker.

A TD time parity error will be forced to cause the logging of a level 1 error during a DMA. The L1 Error Status Register will then be read to check the TRM's ability to place the number of the TIC it is communicating with into this register. TIC bus time outs and internal bus parity errors during the transfer of data will also be tested.

Since the DMA operation cannot be monitored directly, the status of the CCU data area will be used as an indirect indication of the TRM actions.

The DMA Operations test must be run prior to this routine. This routine covers only errors during the transfer of data and addresses between the TRM and CCU. For a complete test of DMA error management the CSCW Change During DMA routine, which covers errors detected while the CSCW is being "built" in the TRM, must also be run.

Commands/Functions Covered:

- AS/DS Timer (for missing DS)
- Data Strobe Checker
- Early termination of operation
- Logging of TIC number in L1 Error Status Register (DMA)

ERC Description

Routine ERC	Meaning	FRU List
TE01	ERROR MANAGEMENT DURING DMA Internal Bus, Even address, HW 1	TRM
0010	No L4 interrupt after DMA	
002x	Unexpected Level 1 interrupt	
0030	Wrong GET CMD Completion data	
0040	Wrong LID value	
0050	Improper setting, Level 2 register	
	TIC Bus Time Out	
0060	No L4 interrupt after DMA	
007x	Unexpected Level 1 interrupt	
0080	Wrong GET CMD Completion data	
0090	Wrong LID value	
00A0	Improper setting, Level 2 register	
	Even address, Odd Address Mode	
00B0	No L4 interrupt after DMA	
00Cx	Unexpected Level 1 interrupt	
00D0	Wrong GET CMD Completion data	
00E0	Wrong LID value	
00F0	Improper setting, Level 2 register	
	Odd address, Even address Mode	
0100	No L4 interrupt after DMA	
011x	Unexpected Level 1 interrupt	
0120	Wrong GET CMD Completion data	
0130	Wrong LID value	
0140	Improper setting, Level 2 register	
	Internal Bus, Odd address, HW 1	
0150	No L4 interrupt after DMA	
016x	Unexpected Level 1 interrupt	
0170	Wrong GET CMD Completion data	
0180	Wrong LID value	
0190	Improper setting, Level 2 register	
	Internal Bus, Odd address, HW 2	
01A0	No L4 interrupt after DMA	
01Bx	Unexpected Level 1 interrupt	
01C0	Wrong GET CMD Completion data	
01D0	Wrong LID value	
01E0	Improper setting, Level 2 register	
	-External Type 1 Error, TIC 1	
01F0	No L4 interrupt after DMA	
0200	Improper setting, Level 2 register	
	-External Type 1 Error, TIC 2	
0210	No L4 interrupt after DMA	
0220	Improper setting, Level 2 register	
	-External Type 1 Error, TIC 3	
0230	No L4 interrupt after DMA	
0240	Improper setting, Level 2 register	
	-TD Bad Parity, TIC 0 DMA WRITE	
0250	No L1 interrupt for parity error	
0260	Improper setting, L1 Status register	
	-TD Bad Parity, TIC 1 DMA WRITE	
0270	No L1 interrupt for parity error	
0280	Improper setting, L1 Status register	
	-TD Bad Parity, TIC 2 DMA WRITE	
0290	No L1 interrupt for parity error	
02A0	Improper setting, L1 Status register	
	-TD Bad Parity, TIC 3, DMA READ	
02B0	No L1 interrupt for parity error	
02C0	Improper setting, L1 Status register	

TE02: MOSS CONTROL BITS (DISCONNECT STATE)

Description

This routine tests the ability of the MOSS CONTROL bits in the TIC Control Register to cause errors to be logged in the MOSS Error Status register and to generate "Direct" interrupts from bits set in the IR Register. The proper logging of errors in the MOSS Error Status register and the correct GET COMMAND COMPLETION responses for the Direct and MOSS Status interrupts is also checked.

For the MOSS Error Status register and MOSS Status interrupt function of the MOSS CONTROL bits, all 3 types of errors (internal, type 1 and type 2) will be forced during both MMIO and DMA operations using TIC position 0. The contents of the MOSS Error Status register and the Command Completion are verified.

Next, a Direct interrupt is generated from each TIC position by setting the corresponding MOSS CONTROL bit and IR bit on (as well as the START bit). The Command Completion is tested.

Commands/Functions Covered:

- Function of MOSS CONTROL bits in TIC Control Register
- MOSS Error Status Register
- Reset of MOSS Error Status Register by GET COMMAND COMPLETION
- GET COMMAND COMPLETION results for Direct Interrupt and MOSS Status Interrupt

ERC Description

Routine ERC	Meaning	FRU List
TE02	MOSS ERROR STATUS/MOSS Control	TRM
	Internal Error During MMIO	
0010	Expected L4 interrupt not received	
0020	Wrong bits in GET CMD Completion	
0030	Wrong setting, MOSS ERR STAT	
0040	MOSS ERR STAT register not reset	
	Type 2 Error During MMIO	
0050	Expected L4 interrupt not received	
0060	Wrong bits in GET CMD Completion	
0070	Wrong setting, MOSS ERR STAT	
0080	MOSS ERR STAT register not reset	
	Type 1 Error During MMIO	
0090	Expected L4 interrupt not received	
00A0	Wrong bits in GET CMD Completion	
00B0	Wrong setting, MOSS ERR STAT	
00C0	MOSS ERR STAT register not reset	
	Internal Error During DMA	
00D0	Expected L4 interrupt not received	
00E0	Wrong bits in GET CMD Completion	
00F0	Wrong setting, MOSS ERR STAT	
0100	MOSS ERR STAT register not reset	
	Type 2 Error During DMA	
0110	Expected L4 interrupt not received	
0120	Wrong bits in GET CMD Completion	
0130	Wrong setting, MOSS ERR STAT	
0140	MOSS ERR STAT register not reset	
	Type 1 Error During DMA	
0150	Expected L4 interrupt not received	
0160	Wrong bits in GET CMD Completion	
0170	Wrong setting, MOSS ERR STAT	
0180	MOSS ERR STAT register not reset	
	Direct Interrupts	
0190	No expected L4, TIC 0 (BR pos)	
01A0	Wrong bits in GET CMD Completion	
01B0	No expected L4, TIC 1	
01C0	Wrong bits in GET CMD Completion	
01D0	No expected L4, TIC 2	
01E0	Wrong bits in GET CMD Completion	
01F0	No expected L4, TIC 3	
0200	Wrong bits in GET CMD Completion	

TE03: BR SCAN WHEEL

Description

This routine tests the ability of the BR scan wheel to service bus requests from the TIC in the proper order. As in the IR Scan Wheel test, patterns of bits are set in the BR register and the order in which the requests are serviced is monitored.

In order to tell which TIC BR is being serviced, an error must be forced during the operation to cause an interrupt to MOSS (internal bus bad parity during the CSCW build will be used to force a CSCW change). The LID will then indicate the serviced TIC.

The DMA operations and CSCW Change routines must be performed before this test.

Commands/Functions Covered:

- BR Scan Wheel

ERC Description

Routine ERC	Meaning	FRU List
TE03	BR SCAN WHEEL	TRM
0010	No L4 after DMA, TIC 0 due	
0020	Wrong LID, TIC 0 not serviced	
0030	No L4 after DMA, TIC 2 due	
0040	Wrong LID, TIC 2 not serviced	
0050	No L4 after DMA, TIC 0 due	
0060	Wrong LID, TIC 0 not serviced	
0070	No L4 after DMA, TIC 1 due	
0080	Wrong LID, TIC 1 not serviced	
0090	No L4 after DMA, TIC 2 due	
00A0	Wrong LID, TIC 2 not serviced	
00B0	No L4 after DMA, TIC 3 due	
00C0	Wrong LID, TIC 3 not serviced	
00D0	No L4 after DMA, TIC 1 due	
00E0	Wrong LID, TIC 1 not serviced	

TE04: CONNECT/DISCONNECT MASK

Description

This routine tests the CONNECT and DISCONNECT operations of the TRM, as well as the function of the MASK command. The TRM is assumed to be in DISCONNECT mode when this routine is run. Interrupts will be generated from each of the TIC positions while in the CONNECT state to verify the operation of the MOSS control bits.

All MOSS CONTROL bits in the TIC Control Register are set on to ensure that any interrupts generated in this procedure are sent to the MOSS. The START (CONNECT) PIO command is issued, and the reset of the DISCONNECT and PIO DISABLE bits in the Level 1 Error Status Register is verified. An interrupt is generated from each of the TIC positions to verify the function of the MOSS CONTROL bits in the CONNECT state.

Now, with the TRA in the CONNECT state, the MASK command is issued. Interrupts are set up using the IR register and Diagnostic Register, but no interrupt to MOSS should be generated by the TRM.

The STOP (DISCONNECT) command is issued, and the proper operations and level 1 register settings are verified (as in routine 2). The MOSS interrupt indicating the completion of the DISCONNECT operation should not be masked. After this interrupt is serviced, no further interrupts should be received.

The UNMASK command is issued and an interrupt is generated to verify its function. The MASK command is issued again, and this time a PROGRAMMED RESET is issued to clear the MASK condition. This is verified by setting up another interrupt through the IR and Diagnostic Registers and servicing it.

As mentioned in routine 2, the function of the bit CCU PIO DISABLE cannot be tested from the MOSS.

Commands/Functions Covered:

- START/STOP
- MASK/UNMASK
- MOSS CONTROL bit function in CONNECT state
- Clearing of MASK by PROGRAMMED RESET
- "Interrupt Masked On" bit of GET COMMAND COMPLETION

ERC Description

Routine ERC	Meaning	FRU List
TE04	CONNECT/DISCONNECT/MASK	TRM
001x	Unexpected interrupt, START command	
0020	Wrong setting, L1 Status register	
	Direct Interrupt, TIC 0	
0030	Expected L4 not received	
0040	Error in GET CMD Completion data	
	Direct Interrupt, TIC 1	
0050	Expected L4 not received	
0060	Error in GET CMD Completion data	
	Direct Interrupt, TIC 2	
0070	Expected L4 not received	
0080	Error in GET CMD Completion data	
	Direct Interrupt, TIC 3	
0090	Expected L4 not received	
00A0	Error in GET CMD Completion data	
00Bx	Unexpected interrupt, MASK command	
00Cx	Interrupt received, not masked	
00Dx	Unexpected interrupt after STOP cmd	
00E0	ADP L1 not set in x'7E' register	
00F0	ADP L1 not reset by GET L1	
0100	No L4 interrupt for end of DISCONNECT	
0110	Error in GCC data	
0120	Wrong setting, L1 Status register	
013x	Interrupt received, not masked	
014x	Unexpected interrupt after UNMASK	
0150	No expected L4, not unmasked	
0160	Error in GCC data	
0170	RESET bit off after PROG RSET	
0180	No expected L4, mask not reset	
0190	Error in GCC data	

TF01: TIC RESET AND INTERNAL TESTS

Description

In this routine, the TIC are reset and the results of their internal tests (run automatically at reset) are obtained. The initialization procedure will then be performed to verify ESS operations (MMIO, and DMA) with the TIC attached.

This procedure exercises most of the MMIO commands and will be tailored to execute all of them. It also exercises the DMA operation.

Problems arising in this routine can be a result of errors in the TIC or the TRM. The procedure must be repeated on each TIC to determine the FRU most likely in error.

All portions of the TRM must have been tested prior to this point.

Commands/Functions Covered:

- All MMIO operations
- DMA operation (to SCB, SSB)
- Path and drivers to/from TIC

ERC Description

Routine ERC	Meaning	FRU List
TF01	TIC RESET AND INTERNAL TESTS	N. A
001x	Unexpected interrupt after clearing wrap mode	
002x	Unexpected interrupt after reset	
0030	Self test time out (retry)	
0040	Hardware error found, self test	
005x	Unexpected interrupt after reset, test	
006x	Interrupt loading initialization parameters	
0070	Error in Autoinc of DATA+ cmd	
0080	Wrong data reading initialization parm1	
0090	Wrong data reading initialization parameters	
00Ax	Interrupt reading initialization parameters	
00B0	Initialization timeout (retry)	
00C0	Hardware error found, initialization phase	
00Dx	Unexpected interrupt, initialization phase	
00E0	Wrong data in SCB	
00F0	Wrong data in SSB	

TF02: TIC BUS PARITY CHECKER

Description

This routine tests the ability of the parity checker (number 2) on the TIC Bus to detect errors in data sent to the TRM from the TIC, and the ability of the TRM to manage the reporting and logging of the error in the proper Level 2 Error Status Register. The TRM must set the correct bits in the response to the GET COMMAND COMPLETION and must calculate the Type B LID for the TIC generating the error.

A TIC will be reset and initialized with the system Parity Test option to cause the TIC to force its parity bits to '0'B for all transfers. The initialization completion is reported as usual, and any parity errors received up to this point are ignored.

Next, specific parity errors are forced to the checker by writing data to the TIC which requires the parity bit to be '1'B and reading it back from the TIC by MMIO. The parity errors will be generated on specific positions: byte 0, byte 1, then both bytes simultaneously. The proper error detection and management is verified for each distinct error.

This test is run using only one of the installed TICs to force parity errors to the TRM.

Commands/Functions Covered:

- TIC Bus parity checker (# 2)

ERC Description

Routine ERC	Meaning	FRU List
TF02	TIC BUS PARITY CHECKER	
001x	Unexpected interrupt after clearing wrap mode	All TICs
0020	Self test time out (retry)	TIC TRM
0030	Hardware error found, self test	
0040	Wrong data reading initialization parameters	
005x	Interrupt during reset, test, load, read	
0060	Initialization timeout (retry)	
0070	Hardware error found, initialization phase	
008x	Unexpected L1 int, initialization phase	
	Error forced on B0, TIC Bus	
009x	Unexpected interrupt on MMIO WRITE	TRM TIC
00A0	No L4 interrupt after MMIO READ	
00Bx	Unexpected L1 on MMIO READ	TRM
00C0	Wrong bits set, GET CMD Completion	
00D0	Wrong LID value	
00E0	Improper logging, L2 Status register	
	Error forced on B1, TIC Bus	
00Fx	Unexpected interrupt on MMIO WRITE	TRM TIC
0100	No L4 interrupt after MMIO READ	
011x	Unexpected L1 on MMIO READ	TRM
0120	Wrong bits set, GET CMD Completion	
0130	Wrong LID value	
0140	Improper logging, L2 Status register	
	Error forced on B0 and B1	
015x	Unexpected interrupt on MMIO WRITE	TRM TIC
0160	No L4 interrupt after MMIO READ	
017x	Unexpected L1 on MMIO READ	TRM
0180	Wrong bits set, GET CMD Completion	
0190	Wrong LID value	
01A0	Improper logging, L2 Status register	

TG01: TIC LOBE TEST/INTERRUPT GENERATION

Description

This routine starts the TIC (internal) Lobe Test and obtains the results of that test. The OPEN command must be issued to the TIC to start the internal Lobe Test, so the SCB/SSB communication and the generation of TIC-to-system interrupts is also tested in this routine.

The TIC is reset and initialized as in "TF01: TIC Reset and Internal Tests", and the OPEN command is issued to the TIC by communication through the TIC Interrupt Register and the SCB. The adapter will be opened with the WRAP option, as this causes the TIC to run only the Lobe Test and not the entire OPEN process. Setting the SCB REQUEST bit in the TIC Interrupt Register causes the TIC to interrupt the system when the SCB is available for another request. The results of the Lobe Test will be placed in the SSB at the completion of the OPEN processing.

After the results of the internal tests have been verified, the CLOSE command will be issued to the TIC, again using the SCB. Another interrupt will be generated when the TIC has cleared the SCB and status will be posted in the SSB following the completion of the command.

The TIC will be reset at the end of the routine to prevent its interference if the routine is to be run on other TICs.

Commands/Functions Covered:

- TIC-to-system interrupt (SCB CLEAR, Command Status)
- TIC internal Lobe Media Test
- SCB/SSB communication mechanism (with TIC Interrupt Register)

ERC Description

Routine ERC	Meaning	FRU List
TG01	TIC OPEN WRAP/LOBE TEST	N. A
001x	Unexpected interrupt after clearing wrap mode	
0020	Self test time out (retry)	
0030	Hardware error found, self test	
0040	Wrong data reading initialization parameters	
005x	Interrupt during reset, test, load, read	
0060	Initialization timeout (retry)	
0070	Hardware error found, initialization phase	
008x	Unexpected interrupt, initialization phase	
0090	Wrong data in SCB	
00A0	Wrong data in SSB	
00B1	No SSB update interrupt after OPEN	
00B2	No SSB update interrupt after CLOSE	
00C1	OPEN status not in SSB	
00C2	CLOSE status not in SSB	
00D1	OPEN error indicated in SSB	
00D2	CLOSE error indicated in SSB	
00E0	Ring stat interrupt, no stat in SSB	
00F0	Ring status indicates error	
0101	No SCB CLEAR interrupt, OPEN	
0102	No SCB CLEAR interrupt, CLOSE	
0111	Wrong vector, SCB CLEAR, OPEN	
0112	Wrong vector, SCB CLEAR, CLOSE	
0121	SCB not cleared, OPEN command	
0122	SCB not cleared, CLOSE command	
013x	Interrupt after SSB update interrupt	

TH01: NON-WRAP DMA ERRORS

Description

This routine tests the management of errors during (non-WRAP mode) DMA operations between a TIC and the TRM.

The first part of the routine tests the generation of the BERR (BUS ERROR) signal to the TIC and the retry of the operation by the TIC. The initialization parameters are set up to allow one DMA retry, and the TIC initialization is begun. The Diagnostic Register is set up to force an internal bus parity error during the DMA performed as the last step of the TIC initialization. A level 4 interrupt should be received and the Level 2 Error Status Register setting is verified. The TIC should retry the DMA operation after receiving the BERR tag, placing the proper data in the SCB and SSB areas in CCU memory.

The TIC is then reset and re-initialized to allow no retry for DMA errors. An error is forced during the DMA of the open command from the SCB.

A pending Adapter Check interrupt from the TIC should be degated by the TRM. This is verified by reading the IR/BR. When the L2 Error Status Register is cleared (by a GET L2 ERROR STATUS), this TIC interrupt should be allowed by the TRM. The contents of the L2 Error Status Register for the DMA error and the TIC Adapter Check interrupt vector are verified.

Commands/Functions Covered:

- BERR to TIC
- DMA retry by TIC
- Degate of IR from TIC for DMA errors
- Generation of Adapter Check interrupt form TIC

ERC Description

Routine ERC	Meaning	FRU List
TH01	NON WRAP DMA ERRORS	N. A
	Generation of BERR to TICs	
001x	Unexpected interrupt after clearing wrap mode	
0020	Self test time out (retry)	
0030	Hardware error found, self test	
0040	Wrong data reading initialization parameters	
005x	Interrupt during reset, test, load, read	
0060	No L4 interrupt for DMA error	
0070	Wrong setting, L2 error status	
0080	Wrong data in SCB	
0090	Wrong data in SSB	
	Degate of IR for DMA Error	
00A0	Self test time out (retry)	
00B0	Hardware error found, self test	
00Cx	Interrupt during reset, test, load	
00D0	Initialization time out	
00E0	Hardware error found, initialization phase	
00Fx	Interrupt during initialization phase	
0100	Wrong data in SCB	
0110	Wrong data in SSB	
0120	No L4 interrupt for DMA error	
0130	Interrupt from TIC not degated	
0140	Wrong setting, L2 error status	
0150	No Adapter Check interrupt	
0160	Wrong interrupt vector in L2 register	
0170	Unexpected interrupt at routine end	

APPENDIX A. ABBREVIATIONS AND GLOSSARY

ABBREVIATIONS

A	ampere
ABEND	abnormal end of task
ac, AC	(1) alternating current (3) address compare
ACB	adapter control block
ACF	Advanced Communication Function
ACK	affirmative acknowledgment (BSC)
ACLK	adapter clock (card)
ACR	add character register (instruction)
ACU	automatic calling unit
AE	address exception
AEK	address exception key
AGC	automatic gain control (signal)
AHR	add halfword register (instruction)
AIO	adapter-initiated operation
AIT	average instruction time
ALU	arithmetic and logic unit
AR	(1) add register (instruction) (2) amplifier
ARC1	auto-restart card in unit 01
ARC2	auto-restart card in unit 02
ARI	add register immediate (instruction)
ASCII	American National Standard Code for Information Interchange
ATTN	attention (3101 operator console key)
B	branch (instruction)
BAL	branch and link (instruction)
BALR	branch and link register (instruction)
BB	branch on bit (instruction)
BCC	block check character (BSC)
BCCW	bit clock control word
BCL	branch on C latch (instruction)
BCT	branch on count (instruction)
BER	box error record
bps	bits per second
BSC	binary synchronous communication
BSM	basic storage module
BSMI	basic storage module interface (card)
BT	branch trace
BTAC	branch trace/address compare (card)
BTAM	basic telecommunication access method
BUSTERM	IOC bus terminator (card)
BZL	branch on Z latch (instruction)
C	(1) Celsius (2) control (X.21 signal)
CA	channel adapter
CAB	channel adapter board
CAC	common adapter code
CADR	channel adapter driver receiver (card)
CADRUK	channel adapter driver receiver type UK (card)
CB	circuit breaker
CCA	communication common adapter (card)
CCIN	channel-to-CCU interface (card)
CCITT	Comite Consultatif International Telegraphe et Telephone
CCLK	CCU clock (card)
CCMD	current command (storage)
CCN	communications controller node
CCR	compare character register (instruction)
CCU	central control unit
CCUB	CCU board
CCW	channel control word

CDF	configuration data file
CDS	configuration data set (NCP/EP)
CE	(1) customer engineer (WTC term for FE) (2) channel end (channel status)
CELIA	CE latched indicator analytic (card)
CHCW	channel control word
CHIN	channel interface (card)
CHR	compare halfword register (instruction)
CLAB	channel and line attachment board
CLDP	controller load/dump program
CNM	communication network management
CNMI	communication network management interface
CNSL	console
CP	command processor
CPA	control panel adapter (card)
CPIT	control progam interface table
CPM	connection point manager
CPT	checkpoint trace
CR	(1) compare register (instruction) (2) call request (signal)
CRC	cyclic redundancy check
CRI	compare register immediate (instruction)
CRP	check record pool
CS	(1) cycle steal (2) communication scanner
CSCW	cycle steal control word
CSG	cycle steal grant
CSGH	cycle steal grant high
CSGL	cycle steal grant low
CSM	communication scanner memory (card)
CSP	communication scanner processor
CSP1	communication scanner processor (card) type 1
CSP2	communication scanner processor (card) type 2
CSR	cycle steal request
CSRH	cycle steal request high
CSRL	cycle steal request low
CSS	control subsystem
CSW	channel status word
CTS	clear to send (signal)
CTL1	control type 1 (card)
CTL2	control type 2 (card)
CVTL	card vendor transistor logic
CZ	Carry/zero (latch)
DAC	hard disk or diskette adapter card
DAF	destination address field (SNA)
DB	data byte (signal)
dc, DC	(1) direct current (2) data chaining (channel status)
DCE	data circuit-terminating equipment
DCF	diagnostic control facility
DCM	diagnostic control monitor
DE	device end (channel status)
DFLx	data flow type x (card) (where x = 1, 4, or 5)
DIFF	differentiator
DLO	data line occupied (signal)
DMA	direct memory access
DP	digit present (signal)
DSx	digit signal 2 to power x
DSC	distant station connected
DSR	data set ready (signal)
DSRS	data signaling rate selection (signal)
DTE	data terminal equipment
DTR	data terminal ready (signal)
DVB	asynchronous-devices (SNA)
DX	duplex (full-duplex)
EBCDIC	extended binary-coded decimal interchange code
EC	engineering change
ECC	error checking and correction (card)
EDE	elementary data exchange
EIA	Electronic Industries Association
ENQ	enquiry (BSC)
EOT	end of transmission (BSC)
EP	emulation program
EPO	emergency power off
ERC	error reference code

EREP	environmental recording, editing, and printing (program)
ERP	error recovery procedure
ESC	emulation subchannel
ESCH	emulation subchannel high
ESCL	emulation subchannel low
ESD	(1) electrostatic discharge (2) external symbol dictionary
ETB	end-of-transmission block character (BSC)
ETX	end-of-text character (BSC)
FAC	flag address control
FCC	Federal Communications Commission
FCPS	final call progress signals (X.21)
FES	front-end scanner (card)
FIC	FRU isolation code
FM	frequency modulation
FNCTN	function (CCU FNCTN) (3101 operator console key)
FPS	FES parameter/status
FRU	field-replaceable unit
ft	foot
GCF	graphic configuration file
GPR	general purpose register
GPT	generalized PIU trace
GTF	generalized trace facility
HDX	half-duplex
HDR1	header 1 (hard disk or diskette)
HLIR	high level interface request
HW	hardware
Hz	Hertz
IAR	instruction address register
IC	insert character (instruction)
ICA	integrated communication adapter
ICB	interface control block (storage)
ICC	internal clock control
ICT	insert character and count (instruction)
ICW	interface control word
ID	identifier (hard disk or diskette)
IFT	internal function test
IMB	in mailbox (MOSS)
IML	initial microcode load(er)
in.	inch
IN	input (instruction)
INN	intermediate network node
INOP	inoperative (line, modem, or terminal)
IOC	input/output control
IOCB	input/output control bus
IOCS	input/output control system
IOH	adapter input/output (instruction)
IOHI	adapter input/output immediate (instruction)
IOIRR	input/output interrupt request register
IML	initial microcode load
INN	intermediate network node
IPF	instruction pre-fetch
IPL	initial program load(er)
IPR	isolated pacing response (SNA)
IRR	interrupt request register
ITB	intermediate text block (BSC)
IVT	isolation verification tests
K	1024 (bytes or words)
KBD	keyboard
kbps	kilobits per second
kg	kilogram
kHz	kilohertz
L	load (instruction)
LA	load address (instruction)
LAB	line attachment board
LABA	line attachment board type A
LABB	line attachment board type B
LABC	line attachment board type C

LAN	local area network
LAR	lagging address register
LCB	line control block (storage)
LCD	line control definer (storage)
LCOR	load character with offset register (instruction)
LCR	load character register (instruction)
LCS	line communication status (storage)
LDF	line description file
LED	light-emitting diode
LH	load halfword (instruction)
LHOR	load halfword with offset register (instruction)
LHR	load halfword register (instruction)
LIB	line interface buffer
LIC	line interface coupler
LIC1	line interface coupler type 1 (card)
LIC2	line interface coupler type 2 (card)
LIC3	line interface coupler type 3 (card)
LIC4A	line interface coupler type 4A (card)
LIC4B	line interface coupler type 4B (card)
LLIR	low level interrupt request
LL2	link level 2 test
LNVT	line vector table (storage)
LOGREC	error logging program of access method
LOR	load with offset register (instruction)
LPDA	link problem determination aid
LR	load register (instruction)
LRC	longitudinal redundancy check
LRI	load register immediate (instruction)
LS	local storage
LSAR	local storage address register
LSI	large scale integration
LSR	local storage register (CSP)
LSSD	level sensitive scan design
m	meter
MAP	maintenance analysis-procedure
MCPC	machine check program check
MCC	MOSS control card
MDOR	MOSS data operand register
MDR	miscellaneous data recorder
MEM	memory (card)
MES	miscellaneous equipment specifications
MFM	modified frequency modulation
MHz	megahertz
MICB	MOSS interface control block
MIM	maintenance information manual
MIO	MOSS input/output
MIOC	MOSS input/output control (card)
MIOH	MOSS input/output halfword
MIOHI	MOSS input/output halfword immediate
MLC	machine level control
MLT	machine load table (hard disk or diskette)
mm	millimeter
MMB	memory and MOSS board
MMC	MOSS memory card
MMM8	MOSS memory module 8K
MMM24	MOSS memory module 24K
MMM32	MOSS memory module 32K
mn	minute
MOD	modifier
MOSS	maintenance and operator subsystem
MPC	MOSS processor card
ms	millisecond
MSA	machine status area (console)
MSD	machine status display
mV	millivolt
NAK	negative acknowledgment (BSC)
NCCF	network communication control facility (CNM)
NCP	network control program
NCR	AND character register (instruction)
NEO	network expansion option
NHR	AND halfword register (instruction)
NLDM	network logical data management
NOSP	network operation support program (VTAM)
NPDA	network problem determination application (CNM)
NR	AND register (instruction)

NRI	AND register immediate (instruction)
NRZI	non return-to-zero inverted
ns	nanosecond
NSC	native subchannel
NTO	network terminal option
NTT	Nippon Telegraph Telephone
oc	overcurrent
OCR	OR character register (instruction)
OEM	original equipment manufacturer
OEMI	original equipment manufacturer's information
OHR	OR halfword register (instruction)
OLTEP	online test execution program
OLTSEP	online test standalone execution program
OLTS	online test system
OLTT	online terminal test
OMB	out mailbox
OP	operation decode
OR	OR register (instruction)
ORI	OR register immediate (instruction)
OUT	output (instruction)
ov	overvoltage
PCF	primary control field (storage)
PCI	program-controlled interrupt
PCR	power check reset
PCW	processor control word
PDF	parallel data field (storage)
PEP	partitioned emulation program
PF	program function (3101 operator console keys)
PFAR	prefetch address register
PH1-x	phase control power block x in unit 01
PH2-x	phase control power block x in unit 02
PH4-x	phase control power block x in unit 02
PIO	program initiated operation
PIRR	program interrupt request register
PN	part number
POPR	prefetch operation register
POR	power-on reset
PS	power supply
PSA	parameter/status area (storage)
PSW	program status word
PTT	post, telephone, and telegraph (agency)
PTX	phototransistor
PV	parity valid (signal)
PWB1	power board unit 01
PWB2	power board in unit 02
PWCA1	power-control analog in unit 01 (card)
PWCA2	power-control analog in unit 02 (card)
PWCL1	power-control logic in unit 01 (card)
PWCL2	power-control logic in unit 02 (card)
PWRC	power resistor card
RA	register to immediate address (instruction)
RAC	repair action code
RAM	random access memory
RAS	reliability, availability, and serviceability
RCAM	RC access method
RCV	receive
RD	receive data (signal)
RDV	redrive (card)
RDVAD	redrive adapter (card)
RE	register external (instruction)
RECFMS	record formatted maintenance statistics
RECMS	record maintenance statistics
REQMS	request for maintenance statistics
RFS	ready for sending (signal) (or clear to send CTS)
RI	register to immediate operand (instruction)
ROK	read-only key
ROS	read-only storage
ROSAR	read-only storage address register
rpm	revolutions per minute
RPO	remote power off
RPQ	request for price quotation

RR	register to register (instruction)
RS	register to storage (instruction)
RSA	register to storage with addition (instruction)
RT	branch (instruction)
RTC	retry count (X.21)
RTM	retry timer (X.21)
RTS	request to send
RVI	reverse interrupt (BSC)
R/W	read/write
s	second
SAR	storage address register
SCB	scanner control block (storage)
SCF	secondary control field (storage)
SCR	(1) subtract character register (instruction) (2) silicon-controlled rectifier
SCTL	storage control (card)
SDF	serial data field (storage)
SDLC	synchronous data link control (SNA)
SE	system engineer
SELN	selection (3101 operator console key)
SES	secondary status (storage)
SHR	subtract halfword register (instruction)
SIO	start input/output
SIT	scanner interface trace
SKDR	storage-protect key data register
SOH	start of heading (BSC)
SP	storage protect
SPAE	storage protect/address exception
SPK	storage protect key
SR	subtract register (instruction)
SRI	subtract register immediate (instruction)
SRL	shift left register
SS	start-stop
SSP	system support programs
ST	store (instruction)
STC	store character (instruction)
STCT	store character and count (instruction)
STH	store halfword (instruction)
STG	storage
STX	start of text (BSC)
SVC	supervisor call
SYN	synchronous idle (BSC)
SYSGEN	system generation
TA	time address
TAP	trace analysis program
TAR	temporary address register
TC	time command
TCAM	telecommunications access method
TCB	task control block
TCC	trace correlation counter (storage)
TCP	test connector pin
TCS	two channel switch (see TPS)
TD	time data
TERMA1	terminator type A in unit 01 (card)
TERMB1	terminator type B in unit 01 (card)
TERM2	terminator in unit 02 (card)
TG	transmission group (NCP line trace)
TIC	token ring interface coupler (card)
TICB	trace interface control block
TIO	test I/O
TPS	two-processor switch (feature) (also referred to as TCS)
TPSA	trace parameter status area
TRA	(1) token ring adapter
TRM	(1) token ring multiplexor (2) test register under mask (instruction)
TRSS	token ring subsystem
TRU	trace record unit
TSET	transmitter signal element timing
TSS	transmission subsystem
TTA	translate table area
TTD	temporary text delay (BSC)
UCW	unit control word
UE	unit exception (channel status)
UEPO	unit emergency power off

UK	United Kingdom
UKA	user key address
UKP	user key program
UKDR	user key data register
UKL	user key level interrupt
USASCII	(see ASCII)
us	microsecond
uv	undervoltage
V	volt
VB	valid byte (signal)
Vac	volts, alternating current
Vdc	volts, direct current
VFO	variable frequency oscillator
VH	valid halfword (signal)
VRC	vertical redundancy check
VTAM	virtual telecommunication access method
V.24	CCITT V.24 recommendation
V.25	CCITT V.25 recommendation
V.28	CCITT V.28 recommendation
V.35	CCITT V.35 recommendation
W	watt
WACK	wait before transmit positive acknowledgment (BSC)
WB	wrap back (signal)
WKR	working register
WSDR	working storage data register
XCR	exclusive OR character register (instruction)
XHR	exclusive OR halfword register (instruction)
XOR	exclusive OR
XR	exclusive OR register (instruction)
XREG	external registers
XRI	exclusive OR register immediate (instruction)
X.21	CCITT X.21 recommendation
YZxxx	EC controlled pages
ZAP	control program modifier function
ZREG	Z register
50PH1	50-Hz phase control in unit 01 (card)
50PH2	50-Hz phase control in unit 02 (card)
60PH1	60-Hz phase control in unit 01 (card)
60PH2	60-Hz phase control in unit 02 (card)

GLOSSARY

This glossary defines all new terms used in this manual. It also includes terms and definitions from the **IBM Vocabulary for Data Processing Telecommunications, and Office Systems,** GC20-1699.

A

access line:

In the hard disk or diskette drive, a line that transmits pulses to turn the stepper motor.

adapter-initiated operation (AIO):

A transfer of up to 256 bytes between an adapter (channel or scanner) and the CCU storage. The transfer is initiated by an IOH/IOHI instruction, and is performed in cycle stealing via the IOC bus.

addressing:

A technique where the control station selects, among the DTEs that share a transmission line, the DTE to which it is going to send a message.

alcohol pad:

A pad soaked with iso-propyl alcohol.

alternate cylinder:

In the hard disk or diskette drive, the area containing sectors that can be assigned in place of sectors that are not usable.

alternate track:

In the hard disk or diskette drive, a track designated to contain data in place of a defective primary track.

asynchronous transmission:

Transmission in which each character is individually synchronized, usually by the use of start and stop elements. The start-stop link protocol, for example, uses asynchronous transmission contrast with 'synchronous transmission.'

auto-answer:

A machine feature that allows a DCE to respond automatically to a call that it receives over a switched line.

auto-call:

A machine feature that allows a DCE to initiate a call automatically over a switched line.

availability:

The degree to which a system or resource is ready when needed to process data.

B

bail assembly:

In the hard disk or diskette drive, a mechanical arm that operates under control of the head load-solenoid to load or release the read/write head load arm.

belt clearance slots:

In the hard disk or diskette drive, grooves in the fan enclosure that permit the ac motor belt to turn without rubbing against the fan enclosure.

binary synchronous communication (BSC):

A uniform procedure, using standardized set of control characters and character sequences, for synchronous transmission of binary-coded data between stations.

box error record (BER):

Information about an error detected by the controller. It is recorded on the hard disk or diskette and can be displayed on the operator console for error analysis.

C

carriage:

In the hard disk or diskette drive, the part that carries the read/write head under control of the stepper motor drive.

central control unit (CCU):

In the 37xx, the controller hardware unit that contains the circuits and data flow paths needed to execute instructions and to control its storage and the attached adapters.

channel adapter (CA):

A communication controller hardware unit used to attach the controller to a host processor.

channel interface:

The interface between the controller and the host processors.

channel and line attachment base (CLAB):

A board that includes the first CAB and LAB of the controller.

collet:

In the hard disk or diskette drive, the part that centers and holds the hard disk or diskette to the drive hub.

common carrier:

In the USA and Canada, a government regulated private company that furnishes the general public with telecommunication service facilities. For example, a telephone or telegraph company (see also "post telephone and telegraph" for countries outside the USA and Canada).

communication controller:

A communication control unit that is controlled by a program stored and executed in the unit. Examples are the IBM 3705 and IBM 3725/3726 and IBM 37xx/37yy.

Communication Network Management (CNM):

An IBM product program that assists the user in identifying network problems from a control point. It is stored in the host processor and comprises the network problem determination application (NPDA) and the network communication control facility (NCCF).

communication scanner:

See 'scanner'.

communication scanner processor (CSP):

The processor of a scanner.

configuration data file (CDF):

A file of the hard disk or diskette that contains a description of all the hardware features (presence, type, address, and characteristics).

control panel:

A panel on the 37xx that contains switches and indicators for the use of the customer's operator and service personnel.

control subsystem (CSS):

The part of the controller that stores and executes the control program, and monitors the data transfers over the channel and transmission interfaces.

customer engineer (CE):

An individual who provides field services for IBM products.

cooling fan:

In the hard disk or diskette drive, a fan that cools the stepper motor.

crosstalk:

In the hard disk or diskette drive, data bits sensed from one track of the hard disk or diskette while the read/write head is reading another track.

cyclic redundancy check (CRC):

A method of error checking performed at the receiving station after a block check character has been received.

D

data circuit-terminating equipment (DCE):

The equipment installed at the user's premises that provides all the functions required to establish, maintain, and terminate a connection, and the signal conversion and coding between the data terminal equipment (DTE) and the line. For example, a modem is a DCE (see "modem".)

Note: The DCE may be separate equipment or an integral part of other equipment.

data terminal equipment (DTE):

That part of a data station that serves as a data source, data sink, or both, and provides for the data communication control function according to protocols. In the 37xx/37yy, the DTE function is achieved by the FES with the associated LIC.

differentiator-amplifier:

An electronic circuit whose output signal is a function of the time rate of change of the input signal.

direct attachment:

The attachment of a DTE to the controller without a DCE.

hard disk or diskette:

A thin, flexible magnetic disk, and its protective jacket, that records the 37xx microcode, diagnostics, error logs, and monitored data.

hard disk or diskette 2D:

A hard disk or diskette used for storing data on both surfaces with twice the usual bit density.

hard disk or diskette drive:

A mechanism that reads and writes hard disk or diskettes.

drive band:

In the hard disk or diskette drive, a metal band connected to the stepper motor pulley and the head carriage assembly.

drive hub:

In the hard disk or diskette drive, a continuously running part that turns the hard disk or diskette at 360 rpm.

duplex transmission:

Data transmission in both directions at the same time. Contrast with 'half duplex.'

E

emulation program (EP):

The function of a network control program to perform activities equivalent to those of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control.

enclosure:

The hard disk or diskette drive motor cooling fan safety cover.

error recovery procedure (ERP):

A procedure designed to help isolate and, where possible, to recover from errors in equipment. The procedures are often used in conjunction with programs that record the statistics of machine malfunctions.

F

front-end scanner (FES):

A circuit that scans the transmission lines, serializes and deserializes the transmitted characters, and manages the line services. It is part of the scanner.

H

half-duplex:

Data transmission in either direction, one direction at a time. Contrast with 'duplex.'

Note: The functional unit using the data circuit determines the choice of direction.

head/carriage:

In the hard disk or diskette drive, the unit that contains the read/write head.

host processor:

(1) A processor that controls all or part of a user application network. (2) In a network, the processing unit in which the

access method for the network resides. (3) In an SNA network, the processing unit that contains a system services control point (SSCP). (4) A processing unit that executes the access method for attached communication controllers. Also called 'host'

I

identifier:

In the hard disk or diskette drive, a character or group of characters used to identify or name an item of data and possibly used to indicate some properties of that data.

initial microcode load (IML):

The process of loading the microcode into a scanner or into MOSS.

initial program load (IPL):

The initialization procedure that causes 37xx control program to commence operation.

input/output control (IOC):

The circuit that controls the input/output from/to the channel adapters and scanners via the IOC bus.

internal clock circuit (ICC):

An optional circuit that provides, through the LICs, the clock control to the DCEs or DTEs that need it.

internal function test (IFT):

A set of diagnostic programs designed and organized to detect and isolate a malfunction.

J

jacket:

A permanently attached cover that protects the hard disk or diskette surface.

L

line:

See 'transmission line'.

line attachment base (LAB):

The unit of modularity of the transmission subsystem. It corresponds to one board and includes mainly the scanners and the line interface couplers.

line interface coupler (LIC):

A circuit that attaches up to four transmission cables to the controller.

Link Problem Determination Aid (LPDA):

A set of test facilities resident in the IBM 386X modems and activated from the control program in the controller.

link protocol:

The set of rules by which a logical data link is established, maintained, and terminated, and by which data is transferred across the link.

longitudinal redundancy check (LRC):

A system of error checking performed at the receiving station after a block check character has been accumulated.

M

maintenance and operator subsystem (MOSS):

The part of the controller that provides operating and servicing facilities to the customer's operator and customer engineer.

microcode:

A program, that is loaded in a processor (for example, the MOSS processor) to replace a hardware function. The microcode is not accessible to the customer.

modem (MOdulator-DEModulator):

A functional unit that transforms logical signals from a DTE into analog signals suitable for transmission over telephone lines (modulation), and conversely (demodulation). A modem is a DCE. It may be integrated in the DTE.

MOSS input/output control (MIOC):

The circuit that controls the input/output from/to the MOSS.

multiplexing:

The division of a transmission facility into two or more channels by allotting the common channel to several different channels, one at a time.

multipoint connection:

A connection established among more than two data stations for data transmission. The connection may include switching facilities.

N

network:

See 'user application network'.

Network Control Program (NCP):

A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

nonswitched line:

A permanent dedicated transmission line that connects two or more DTEs. The connection can be point-to-point or multipoint. The line can be leased or private. Contrast with 'switched line.'

O

online tests:

Testing of a remote data station concurrently with the execution of the user's programs (that is, with only minimal effect on the user's normal operation).

operator console:

The IBM 3101 Operator Console that is used to operate and service the 37xx through the MOSS. A primary operator console must be located within 5 m (16 ft) of the 37xx. Optionally an alternate operator console may be installed up to 150 m (492 ft) from the 37xx.

P

partitioned emulation programming (PEP):

A feature of NCP that permits some lines to operate in network control mode while simultaneously operating others in emulation mode.

phototransistor:

An electronic part used to sense the light of a light-emitting diode.

point-to-point connection:

A connection established between two data stations for data transmission. The connection may include switching facilities.

polling:

The process whereby stations are invited, one at a time, to transmit.

post telephone and telegraph (PTT):

A generic term for the government-operated common carriers in countries other than the USA and Canada. Examples of the PTT are the Post Office Corporation in the United Kingdom, the Deutsche Bundespost in Germany, and the Nippon Telephone and Telegraph Public Corporation in Japan.

program-initiated operation (PIO):

A transfer of four bytes between a general register in the CCU and an adapter (channel or scanner). The transfer is initiated by IOH/IOHI instruction and is executed via the IOC bus.

R

redrive card:

A card that repowers the IOC bus signals at board entry. It also has logical and checking functions.

reliability:

The ability of a functional unit to perform its intended function under stated conditions, for a stated period of time.

S

scanner:

A device that scans and controls the transmission lines. It is composed of one communication scanner processor (CSP) and one front-end scanner (FES).

services:

A set of functions designed to facilitate the maintenance of a device or system.

serviceability:

The capability to perform effective problem determination, diagnosis, and repair on a data processing system.

solenoid plunger:

In the hard disk or diskette drive, a moving part of the solenoid that operates the bail assembly to load and release the read/write head load arm.

start-stop:

A data transmission system in which each character is preceded by a start signal and is followed by a stop signal.

stepper motor:

In the hard disk or diskette drive, the motor that steps the head carriage assembly from track to track.

switched line:

A transmission line with which the connections are established by dialing, only when data transmission is needed. The connection is point-to-point and uses a different transmission line each time it is established. Contrast with 'nonswitched line.'

synchronous data link control (SDLC):

A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SLDC conforms to subsets of the Advanced Data Communication Control Procedures of the American National Standards Institute and High-level Data Link Control (HDLC) of the International Standards Organization.

synchronous transmission:

Data transmission in which the sending and receiving instruments are operating continuously at substantially the same frequency and are maintained, by means of correction, in a desired phase relationship. Contrast with 'asynchronous transmission.'

systems network architecture (SNA):

The description of the logical structure, formats, protocols, and operational sequences for transmitting information through a user application network. The structure of SNA allows the users to be independent of specific telecommunication facilities.

T

timeout:

The time interval allotted for certain operations to occur.

transmission interface:

The interface between the controller and the user application network.

transmission line:

The physical means for connecting two or more DTEs (via DCEs). It can be nonswitched or switched. Also called a 'line.'

transmission subsystem (TSS):

The part of the controller that controls the data transfers over the transmission interface.

tunnel erase circuit:

In the hard disk or diskette drive, an electronic circuit that is used to erase the edge of the track just recorded during a

write operation. This erasing prevents crosstalk between track during later read operations.

two-processor switch (TPS):

A feature of the channel adapter that connects a second channel to the same adapter.

U

user application network:

A configuration of data processing products, such as processors, controllers, and terminals, for the purpose of data processing and information exchange. This configuration may use circuit-switched, packet-switched, and leased-circuit services provided by carriers or PTT. Also called a 'user network.'

V

variable frequency oscillator:

An electronic circuit that is used to synchronize the MOSS reading circuits with the hard disk or diskette drive when it is performing a read operation.

vertical redundancy check (VRC):

An odd parity check performed on each character of a block as the block is received.

W

write/erase:

Writing data to and erasing from a hard disk or diskette.

3725/3726 Communication Controller
and Expansion
Diagnostic Descriptions
Order No. SY33-2027-4

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for customer engineers. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language: English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the branch office serving your locality.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

. . . . . .
. . . . . .
. . . . . .
. . . . . .

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

What is your occupation? . . . . . .

Number of latest Newsletter associated with this publication: . . . . . .

Thank you for your cooperation. No postage stamp necessary if mailed in the USA. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1LG
180 Kost Road
Mechanicsburg
PA 17055

Fold and tape | Please Do Not Staple | Fold and tape

IBM

Cut or Fold Along Line

3725/3726 Communication Controller and Expansion Diagnostic Descriptions Printed in USA SY33-2027-4

SY33-2027-4

3725/3726 Communication Controller and Expansion Diagnostic Descriptions Printed in USA SY33-2027-4

IBM®

SY33-2027-04